ABSTRACT

SYSTEMS SCIENCE (ENGINEERING CONCENTRATION)

FADIRAN, OLADIPO O. B.ENGR. UNIVERSITY OF ILORIN, 1997

M.Sc. UNIVERSITY OF CAPE TOWN, 2001

ADAPTIVE SAMPLING BY HISTOGRAM EQUALIZATION:

THEORY, ALGORITHMS, AND APPLICATIONS

Advisor: Dr. Péter Molnár
Dissertation dated May 2007

We present the investigation of a novel, progressive, adaptive sampling scheme. This scheme is based on the distribution of already obtained samples.

Even spaced sampling of a function with varying slopes or degrees of complexity yields relatively fewer samples from the regions of higher slopes. Hence, a distribution of these samples will exhibit a relatively lower representation of the function values from regions of higher complexity. When compared to even spaced sampling, a scheme that attempts to progressively equalize the histogram of the function values results in a higher concentration of samples in regions of higher complexity. This is a more efficient distribution of sample points, hence the term adaptive sampling. This conjecture is confirmed by numerous examples.

Compared to existing adaptive sampling schemes, our approach has the unique ability to efficiently obtain expensive samples from a space with no prior knowledge of the relative levels of variation or complexity in the sampled function. This is a requirement in numerous scientific computing applications.

Three models are employed to achieve the equalization in the distribution of sampled function values: (1) an active-walker model, containing elements of the random walk theory, and the motion of Brownian particles, (2) an ant model, based on the simulation of the behavior of ants in search of resources, and (3) an evolutionary algorithm model. Their performances are compared on objective basis such as entropy measure of information, and the Nyquist-Shannon minimum sampling rate for band-limited signals.

The development of this adaptive sampling scheme was informed by a need to efficiently synthesize hyperspectral images used in place of real images. The performance of the adaptive sampling scheme as an aid to the image synthesis process is evaluated. The synthesized images are used in the development of a measure of clutter in hyperspectral images. This process is described, and the results are presented.

ADAPTIVE SAMPLING BY HISTOGRAM EQUALIZATION: THEORY, ALGORITHMS, AND APPLICATIONS

A DISSERTATION
SUBMITTED TO THE SCHOOL OF ARTS AND SCIENCES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY

BY

OLADIPO O. FADIRAN

DEPARTMENT OF ENGINEERING, SYSTEM SCIENCE PROGRAM

ATLANTA, GEORGIA

MAY 2007

R = xii T = 162

ACKNOWLEDGMENTS

I appreciate the immense contributions of Dr. Péter Molnár, my advisor, and Dr. Lance Kaplan, my co-advisor. They provided direction, encouragement, and constructive criticisms, all in the right proportions. It has been great working with you. I am also grateful to the other members of my dissertation committee, namely, Dr. George Japaridze, Dr. William Seffens, and Dr. BhuDev Sharma. Their various expertise formed an invaluable pool of resources that was always available to me.

Also worthy of mention is Dr. Ayodele Abatan, who along with my co-advisor facilitated my enrollment in the Ph.D program. The Systems Science program coordinator, Dr. Lebone Moeti, was always accessible and quick to address any issues. I am grateful to Dr. Alfred Msezane and the other members of the Center for Theoretical Physics for creating a stimulating environment for research work.

I would also like to express my gratitude for the financial support that I received from the United States Army research laboratory. My research work was fully funded through the "Science of Land Target Signatures" project underwritten by this agency.

I am grateful to family members and friends who contributed to making this happen. I am grateful to God for putting you in my life.

To Temitope(iyebiye mi), Babayosimi, and *Atinuke*. This would not have been possible without your unwavering support through thick and thin. I am truly blessed to have you.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ii

LIST OF FIGURES vii

LIST OF TABLES xi

Chapter Page

1 INTRODUCTION 1

1.1 Background and Motivation 1

1.1.1 Adaptive Sampling by Histogram Equalization (ASHE) . . 1

1.1.2 Model for Quantifying Clutter in Hyperspectral Images . . 6

1.2 Dissertation Outline 8

2 ADAPTIVE SAMPLING BY HISTOGRAM EQUALIZATION (ASHE) ALGORITHM 10

2.1 Review of Adaptive Sampling Algorithms 10

2.2 Theoretical Basis 13

2.3 Illustration of ASHE 16

2.4 Application Areas 20

2.4.1 Data Synthesis 20

2.4.2 Design of Experiments 21

2.4.3 Surface Reconstruction from Expensive Samples 21

2.4.4 Progressive Transmission/Rendering 22

3 MODELS UTILIZED IN IMPLEMENTING ASHE 23

3.1 Active Walker Model 23

3.1.1 ASHE Implementation using Active Walkers 25

3.2 Ant Model 29

TABLE OF CONTENTS

Chapter			Page
		3.2.1 ASHE Implementation using the Ant Model	31
	3.3	Evolutionary Algorithm Model	34
		3.3.1 ASHE Implementation using the Evolutionary Algorithm Model	34
4	PERFORMANCE AND SENSITIVITY ANALYSIS OF MODELS		38
	4.1	Measure Based on Frequency Content	39
	4.2	Measure Based on Entropy Measure of Information	44
	4.3	Analysis of the Active Walker Model	44
	4.4	Analysis of the Ant Model	56
	4.5	Analysis of the Evolutionary Algorithm (EA) Model	73
	4.6	Further Analysis of the Active Walker Model	84
	4.7	Conclusions	85
5	NATURE, USES AND SYNTHESIS OF HYPERSPECTRAL IMAGES		89
	5.1	Nature of Hyperspectral Images	89
	5.2	Synthesizing Hyperspectral Images	91
		5.2.1 The Digital Imaging and Remote Sensing Image Generation model (DIRSIG)	92
	5.3	Image Synthesis with DIRSIG	93
6	EFFICIENT HYPERSPECTRAL IMAGE SYNTHESIS USING ASHE		100
	6.1	ASHE based Image Synthesis	101
		6.1.1 Imaged Scene	102
		6.1.2 Input Factors to DIRSIG	103

TABLE OF CONTENTS

Chapter Page

6.1.3 Baseline ATR Performance 103

6.1.4 Image Synthesis 105

6.2 Experiments 105

7 DEVELOPMENT OF A MEASURE OF CLUTTER FOR HYPERSPECTRAL IMAGES 110

7.1 Clutter Complexity Measure 111

7.2 Clutter Complexity Measure for Single Hyperspectral Bands using Real Data 112

7.2.1 Baseline ATR Performance 113

7.2.2 Multiple-feature CCM 115

7.2.3 Single-feature CCM 116

7.2.4 Single-Band CCM Experiments 117

7.3 Clutter Complexity Measure for Hyperspectral Images 121

7.3.1 Baseline ATR Performance 121

7.3.2 Image Clutter Metrics 123

7.3.3 Determining Significant Metrics 124

7.3.4 Hyperspectral Images CCM Experiments 129

8 CONCLUSIONS 138

8.1 Summary of contributions 139

8.2 Suggestions for further work 142

A Clutter Complexity Metrics 144

A.1 Single-band clutter metrics 144

A.1.1 Global standard deviation 144

TABLE OF CONTENTS

Chapter Page

A.1.2 Schmieder Weathersby 144

A.1.3 Homogeneity 145

A.1.4 Energy 145

A.1.5 Entropy 145

A.1.6 Target Interference Ratio 145

A.1.7 Outlier/Edge 146

A.1.8 FBM Hurst Parameter 146

A.1.9 Metrics c and p derived from Gabor filtered images [9] . . 147

A.2 Metric derived from band information content 148

A.2.1 Band correlation 148

A.3 Anomaly detectors 148

A.3.1 Dot product 148

A.3.2 Kullback-Leibler 148

A.4 Metrics derived from the Gray Level Co-occurrence Matrix 149

LIST OF FIGURES

Figure		Page
1.1	Basis of Adaptive Sampling by Histogram Equalization (ASHE)	4
2.1	Illustration of basis for the ASHE algorithm	14
2.2	Illustration of Performance of ASHE, in sampling 2-dimensional function	18
2.3	Illustration of Performance of ASHE, in sampling 2-dimensional image	19
3.1	Simple symmetric random walk on $\mathbb{Z}^2$	24
3.2	Illustration of 2 possible paths for ants to reach a resource. With path 'a' shorter than 'b'	30
3.3	Illustration of standard genetic algorithm recombination process.	35
4.1	Images with single spatial frequencies, and their magnitude images in the frequency domain	40
4.2	Cross-sections of images in frequency domain	41
4.3	Frequency content based measure of function complexity	43
4.4	Entropy based measure of function complexity	45
4.5	Two test functions: smooth, and rapidly varying	46
4.6	Intermediate steps from sampling a smooth test function with the active walker model	49
4.7	Intermediate steps from sampling a rapidly varying test function with the active walker model	50

LIST OF FIGURES

Figure Page

4.8 Intermediate steps from sampling a smooth test function with the active walker model, resulting in good performance. 57
4.9 Intermediate steps from sampling a smooth test function with the active walker model, resulting in poor performance. 58
4.10 Intermediate steps from sampling a rapidly varying test function with the active walker model, resulting in good performance. 59
4.11 Intermediate steps from sampling a rapidly varying test function with the active walker model, resulting in poor performance. 60
4.12 Model to simulate reduction of pheromone concentration as a function of the distance away from source. Concentration $\propto 1/d^2$. 61
4.13 Pheromone concentration/probability change from intermediate steps in the sampling of a smooth function using the ant model 62
4.14 Pheromone concentration/probability change from intermediate steps in the sampling of a rapidly varying function using the ant model 63
4.15 Intermediate stages from sampling a smooth test function with the ant model, resulting in good performance 69
4.16 All stages from sampling a smooth test function with the ant model, resulting in poor performance 70
4.17 Intermediate stages from sampling a rapidly varying test function with the ant model, resulting in good performance 71
4.18 All stages from sampling a smooth test function with the ant model, resulting in poor performance 72

LIST OF FIGURES

Figure Page

4.19 Intermediate stages from sampling a smooth test function with the Evolutionary Algorithm model, resulting in good performance. 79

4.20 All stages from sampling a smooth test function with the Evolutionary Algorithm model, resulting in poor performance. 80

4.21 Intermediate stages from sampling a rapidly varying test function with the Evolutionary Algorithm model, resulting in good performance. . . . 81

4.22 All stages from sampling a rapidly varying test function with the Evolutionary Algorithm model, resulting in poor performance. 82

4.23 *Slice* from 3-dimensional Sinc test function. (a) top view, (b) side view. 84

5.1 Example hyperspectral image, and material signatures. 90

5.2 Combination of factors used to generate images in synthesized database. Total of 216 images. 95

5.3 Example hyperspectral image bands from the same scene 96

5.4 Example hyperspectral images bands from different scenes 97

5.5 Example hyperspectral scenes showing levels of clutter. 98

5.6 Example spectral signatures of materials in synthesized scenes 99

6.1 A single band from the hyperspectral image of the urban scene used for synthesizing multiple images. 102

6.2 Distribution of baseline ATR performance values for different sampling methods 108

LIST OF FIGURES

Figure Page

7.1 RX detection in FLIR images: (a) Original image (b) Image of RX statistic (c) Resulting detection image. 114

7.2 Scatter plot of clutter complexity measure and false alarm count: (a) Weighted sum of multiple features (b) Single feature. 117

7.3 FLIR band classification by clutter complexity measure 118

7.4 Example of band selection based on wavelength to ensure a uniform distribution of the choice of hyperspectral bands. 119

7.5 Performance of clutter complexity measures represented as average false alarm count versus number of bands integrated into the RX detector. . . 120

7.6 Target detection in HSI using the Adaptive Coherence Estimator (ACE) detector 122

7.7 Gray Level Co-occurrence Matrices from hyperspectral images 125

7.8 Gabor filtered band from hyperspectral image 126

7.9 Frequency of selection of clutter metrics for different training image sets 131

7.10 Distribution of CC values resulting in the average values shown in Table7.4(a). The indicated percentages, and the actual number of images that they represent are also shown in the same table. 136

LIST OF TABLES

Table		Page
2.1	Comparison of the Requirements for the Proposed ASHE Algorithm to other Adaptive Sampling Methods.	12
4.1	Performance of active walker model in sampling the smooth function, sorted by the frequency based measure.	51
4.2	Performance of active walker model in sampling the smooth function, sorted by the entropy based measure.	52
4.3	Performance of active walker model in sampling the rapidly varying function, sorted by the frequency based measure.	53
4.4	Performance of active walker model in sampling the rapidly varying function, sorted by the entropy based measure.	54
4.5	Performance of the ant model in sampling the smooth function, sorted by the frequency based measure.	65
4.6	Performance of the ant model in sampling the smooth function, sorted by the entropy based measure.	66
4.7	Performance of the ant model in sampling the rapidly varying function, sorted by the frequency based measure.	67
4.8	Performance of the ant model in sampling the rapidly varying function, sorted by the entropy based measure.	68

LIST OF TABLES

Table Page

4.9 Performance of the Evolutionary Algorithm model in sampling the smooth function, sorted by the frequency based measure. 75

4.10 Performance of the Evolutionary Algorithm model in sampling the smooth function, sorted by the Entropy based measure. 76

4.11 Performance of the Evolutionary Algorithm model in sampling the rapidly varying function, sorted by the frequency based measure. 77

4.12 Performance of the Evolutionary Algorithm model in sampling the rapidly varying function, sorted by the entropy based measure. 78

4.13 A summary of the results from the analysis of the three sampling models. It shows the ranges of input factors that result in good performance, and averages of the entropy based performance values. 83

4.14 Performance of Active Walker model in sampling the 3-dimensional Sinc function, sorted by the frequency based measure. 86

4.15 Performance of Active Walker model in sampling the 3-dimensional Sinc function, sorted by the entropy based measure. 87

7.1 List of some of the image statistical features used in deriving the clutter complexity measure. 115

7.2 Summarized list of clutter metrics used in deriving the clutter complexity measure for hyperspectral images. 127

7.3 Averaged correlation coefficients obtained between the clutter measure and false alarm rates using different training sample sizes. 132

7.4 Averaged correlation coefficients obtained between the clutter measure and false alarm rates using different training sample sizes. 134

CHAPTER 1

INTRODUCTION

The main contribution of this work is the development of a novel, progressive, adaptive sampling method, based on the distribution of already obtained samples. This algorithm is shown to be efficient for sampling a space for which there is no prior information on the global, and relative levels of local variation of the function in question, and the cost of obtaining each sample is prohibitive. This is a requirement in numerous scientific computing applications.

We present results of applying the developed algorithm in the efficient synthesis of hyperspectral images. These images are used in the development of a framework for quantifying clutter in hyperspectral images.

1.1 Background and Motivation

1.1.1 Adaptive Sampling by Histogram Equalization (ASHE)

The reconstruction of most continuous functions from a finite number of sample points results in errors. Since there is always a constraint on the number of samples that can be obtained, the aim of efficient sampling schemes is to minimize the inherent errors that result from reconstructing a continuous function from the finite discrete samples.

An alternate approach to the sampling question is based on the Nyquist-Shannon minimum sampling theory. This shows that a sampling rate N of at least twice the highest frequency component f_s in a signal is required in order to unambiguously reconstruct the

signal from its samples. Thus, the required sampling rate is

$$N \geq 2 \times f_s \,. \tag{1.1}$$

That is, a higher sampling rate is required to unambiguously reconstruct a function with higher frequency components. Reconstructing a continuous signal from finite samples is equivalent to representing finite frequency components of the signal. Error-free reconstruction of a continuous function that is not bandlimited, requires the representation of infinite frequency components [43, 56, 67].

Some sampling algorithms focus more on reducing the *effect* of this inherent error, as is the case in image processing, where structured, and thus more apparent artifacts like aliasing are converted to noise [13, 15, 60]. Others attempt to reduce the actual error by distributing the limited samples more efficiently [52, 53, 64, 71]. A third group of algorithms combines efficient distribution of the samples with the reduction of the effect of the error, as can be found in the adaptive variant of the algorithm discussed here [22]. Algorithms that distribute samples efficiently usually harness the nonstationary nature of the function[1] to be sampled [65]. That is, samples are distributed based on local statistics. This information may be required prior to the sampling process [22], or continuously made available and updated during the sampling process, using a progressive sampling approach [64, 71].

For actual error reduction, the problem of allocating sample points efficiently becomes trivial if there is prior knowledge of levels of variation or local frequency components in a function. The samples are simply allocated based on the different levels of variation. This means that the regions of rapid variation or higher frequencies are allo-

[1] Statistical properties of nonstationary functions, such as the mean change over time

cated relatively more samples. In many cases, there is no *a priori* information on the global, or relative levels of local variation of the function being sampled. Without such knowledge, sample points are usually placed randomly, spaced evenly, or some other variant of these arrangements to avoid the artifact resulting from even spacing [13, 15]. These approaches are however, inefficient for sampling a nonstationary function.

One approach to solving this problem is the progressive intensification of sampling in a local region based on some information content criterion as in the work on ray-tracing [52, 71]. Variable sampling rates may also be achieved using variants of Markov-chain Monte-Carlo (MCMC) methods adapted for this purpose [16, 41]. Local sampling rates may also be pre-determined based on prior information on the local complexities in the function to be sampled. An example of this is found in the adaptive form of the farthest point algorithm [22]. These and similar methods however, require at least one of the following: *a priori* knowledge on the global, or relative levels of local variation of the function to be sampled [21, 22], computation to determine local information content [52, 71, 88], or an acceptance/rejection step in the progressive sampling process [16, 41]. These requirements make these methods infeasible for sampling in many applications.

We present a progressive adaptive sampling algorithm, in which the subsequent sample locations are determined based on the distribution of already collected samples. The algorithm is based on the thesis that even spaced sampling of a function with varying degrees of complexity results in a distribution of samples with relatively lower representation of values from regions of higher complexity. A simple illustration of this can be seen in Figure 1.1. Since the slope in part I of the function is higher than that in part II, more samples are collected per unit length of the function in part II. This is reflected in the distribution of the function values. It is more efficient to concentrate more samples in the region of higher slope or complexity. This results in a reduction of the dominance of

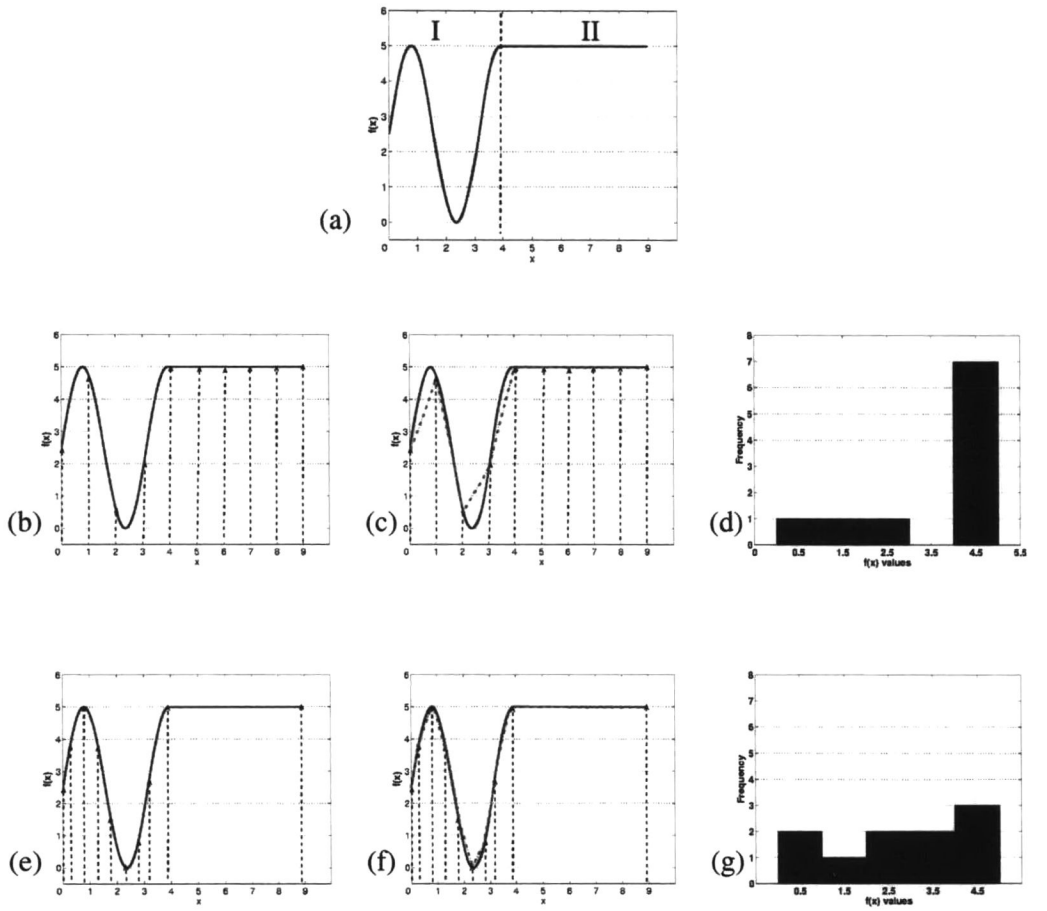


Figure 1.1: This figure illustrates the basis of adaptive sampling by histogram equalization. Figure (a) shows the function to be sampled, (b) shows an evenly spaced set of 10 samples with its linear reconstruction depicted in (c). Figure (d) is the histogram of resulting function values from evenly spaced samples. Figures (e)-(g) show the corresponding results for adaptively placed samples. The error of a linear reconstruction based on the sampled points, represented by the dotted lines, is significantly greater for the even spaced sampling, (c), than for the adaptively placed samples, (f).

the function values from the regions of lower slope, and relatively more samples in the other region. That is, a distribution of function values that tends more towards a uniform distribution. The improved efficiency in sample distribution is evidenced by the function reconstruction based on the samples, indicated by the dotted lines. The concept of samples per unit length, which this example is based on, can be easily extended to higher dimensions.

A sampling scheme that progressively attempts to equalize the histogram of these function values results in a relatively higher concentration of samples in regions of complexity. This results in a more efficient distribution of sample points, hence the adaptive sampling. This conjecture is confirmed by numerous examples shown in Chapter 3 of this dissertation. We call the algorithm Adaptive Sampling by Histogram Equalization (ASHE). The algorithm is not subject to the limitations of the adaptive algorithms mentioned earlier, and only requires that it is possible to obtain the value of the function at each sampled point. No prior knowledge of the local or global levels of variations in the function is required. Also, the only extra computational overhead required by this algorithm is the computation of a histogram at each stage of the sampling procedure. Finally, there is no acceptance/rejection step in the progressive sampling procedure, every obtained sample is kept. This makes the procedure particularly useful for obtaining expensive samples[2].

Three models are employed to achieve the progressive equalization in the distribution of sampled function values. These are:

1. an active-walker model [46, 48, 49], with basis in both the random walk theory [72], and the motion of Brownian particles [75],

[2]Situations for which obtaining each sample is prohibitive in cost, time or some other resource. Good examples are ab initio computations in the physical sciences.

2. an ant model, based on the simulation of the behavior of ants in search of resources [18, 19, 87], and

3. an evolutionary algorithm [61] model.

These are evaluated on their ability to achieve our objective of efficient sample distribution. They are also compared on the basis of ease of implementation. Appropriate models for specific applications are identified based on the analysis of these results.

1.1.2 Model for Quantifying Clutter in Hyperspectral Images

A specific application of the adaptive sampling scheme reported in this work is the efficient synthesis of images. In fact, the development of the adaptive sampling algorithm was informed by the need for such a scheme in this application. The synthesized images are used in our framework for modeling clutter in images. Models of targets and clutter, aid in the understanding of images in general [85, 94]. Targets are considered to be objects of interest in a particular image. We define clutter as any factor in the image that may increase the difficulty for an Automatic Target Recognition (ATR) algorithm in detecting or identifying a target in a scene [23, 25, 27].

Our objective is to obtain a measure of the amount of clutter in an image that will be an indication of the inherent difficulty for an ATR to find a target. This measure will form bounds on the performance of any ATR, such that a high value of this measure will indicate that an ATR will produce a high false alarm (FA) rate. A low value may, however, not result in a low FA rate. This depends on the exact nature of the ATR. Such a measure could serve as a basis for evaluating, and comparing ATRs on an objective basis. It could also serve as a basis for measuring image quality, independent of a particular target detection algorithm or scheme [66].

Our approach to obtaining a clutter measure in these images is to compute a set of statistical image features that are significant for, and monotonically related to ATR performance. In addition, these features have to be algorithmically uncomplicated to implement [27, 66]. The measure of clutter is then obtained as an aggregation of these features that correlates best with baseline ATR performance. The process of combining these features to yield the required result is obtained through a training process on a subset of available image data. Once established, this is generalized over the complete dataset [23, 25, 27].

This training process requires image data in numbers that are statistically significant. There is limited availability of these in the public domain. This limitation can be overcome by synthesizing the desired images. Tools for such image synthesis require inputs like object and scene geometry, object material properties, atmospheric conditions, and illuminating sources [10, 76]. These factors are then accounted for in the ray-tracing process that produces the final image. A database of images can then be produced by synthesizing images with a varying combination of these inputs.

To ensure that the result of the clutter analysis of images from such a synthetic database is general, and representative of real images, two basic requirements should be met. These are the fidelity of each image, and representation in all categories of ATR difficulty in the database. The former requirement is beyond the scope of this work. To achieve the second requirement, an image is modeled as a point on a multidimensional surface, whereby each dimension represents an input parameter to the image synthesis software. Thus, each image results from synthesizing with a combination of input parameters, with each being a possible source of variation with respect to ATR performance. The aim is to sample this surface in order that the resulting images show adequate statistical representation for all categories of ATR difficulty. The prohibitive cost of image

synthesis places a limitation on the number of samples that can be produced [10, 76]. There is also no prior knowledge of how the ATR performance varies with changes in the input parameters used to synthesize the images. Relatively more dense sampling in the regions of this space with higher variability with respect to ATR performance results in a more diverse set of images, and vice-versa [24, 26]. We sample this surface using the ASHE algorithm, and investigate the improvement in performance with respect to diversity in the synthesized images.

1.2 Dissertation Outline

In Chapter 2, we review some existing adaptive sampling schemes and summarize some of their limitations, especially those addressed by our proposed sampling scheme. We then establish the premise on which the Adaptive Sampling by Histogram Equalization (ASHE) algorithm is based. This is done through graphical and analytical methods. The improvement of sampling by the ASHE algorithm over random or even spaced sampling is demonstrated through examples illustrating its performance.

Three stochastic optimization models employed in implementing the ASHE algorithm are discussed in Chapter 3. The underlying theory and heuristics that these models are based on are discussed. We then describe the specific adaptation of the general forms for our particular application.

A performance and sensitivity analysis of the three models is carried out in Chapter 4. We establish two performance measures based on the entropy measure of information [34, 82], and the Nyquist-Shannon minimum sampling rate for band-limited signals [43, 67]. These give an indication of the level of variation or complexity in a function. The measure is computed as the correlation between these two measures separately, and the sample density distribution obtained by employing each model. Hereby,

a higher correlation indicates better performance. We conduct a sensitivity analysis of each model to determine the change in performance for various input factors to each model. This analysis is similar to previous work in ant models [2]. The results obtained from these, serve as good indicators of the appropriateness of each model for particular applications.

The second part of the dissertation, comprising of Chapters 5 - 7, presents results from an application that utilizes synthesized hyperspectral images. An important step in this application is aided by the ASHE algorithm. Chapter 5 gives a background on the nature and uses of hyperspectral images. The need for image synthesis is stated, and the process is described.

The need for an adaptive sampling scheme in the image synthesis process is identified in Chapter 6. The properties of the ASHE algorithm are reiterated to justify its choice for this application. Images are then synthesized based on the ASHE algorithm, and the recorded performance improvement is evaluated on objective basis.

A framework for modeling clutter in hyperspectral images is detailed in Chapter 7. The process of quantifying clutter using both real and synthesized images is then described. Numerous experiments are carried out to investigate this framework.

Chapter 8 concludes with a summary of the findings, and major contributions of the dissertation. Suggestions for future work are also made.

CHAPTER 2

ADAPTIVE SAMPLING BY HISTOGRAM EQUALIZATION (ASHE) ALGORITHM

In this chapter, we propose a novel, progressive, adaptive sampling scheme, based on the distribution of obtained samples. We conduct a brief review of some existing adaptive sampling schemes, and compare them to the proposed algorithm. This is in order to highlight the limitations of the existing algorithms that are addressed by Adaptive Sampling by Histogram Equalization (ASHE). Next, we layout the theoretical basis for the adaptive sampling algorithm. We employ analytical and graphical methods to illustrate why, and how the ASHE algorithm works. Examples are presented to illustrate the performance of the algorithm. Finally, we identify possible practical applications areas of the algorithm.

2.1 Review of Adaptive Sampling Algorithms

Processes like the reconstruction of continuous signals from finite samples, and numerical integration of a continuous signal will generally result in error. This is because they attempt to represent a continuum with a discrete space. These errors can be reduced by using relatively more samples in intervals where the sampled values vary the most. This is the aim of adaptive sampling. That is, the efficient distribution of a finite number of samples in a manner that reflects the varying levels of rate of change, or complexity in a sampled function, in order to minimize errors. Even spaced sampling of a function with nonstationary statistics is inefficient. Adaptive sampling places relatively more samples in regions of higher variance in samples.

An approach to solving this problem is the progressive intensification of sampling in a local region based on some information content criterion as in these works on ray-tracing found in [64, 71]. These algorithms utilize a refinement scheme to determine where to increase the sample density. For example, the variance of sample values in a region is computed, and the sampling density is increased in that region until a threshold is reached [52]. Other measures, such as contrast have also been used as the basis for further refinement [60]. Variable sampling rates may also be achieved using variants of Markov Chain Monte-Carlo (MCMC) methods adapted for this purpose as discussed in [41, 16]. Local sampling rates may also be pre-determined based on prior information on the local complexities in the function to be sampled. An example of this is found in the adaptive form of the farthest point algorithm discussed in [22]. The gradient based sampling algorithms increase sampling density in a region where the slope in the measured quantity exceeds a set threshold. These, and similar methods however, require at least one of the following: *a priori* knowledge on the global, and relative level of local variation of the function to be sampled [22], computation to determine local information content [71, 64, 60], an acceptance/rejection step in the progressive sampling process [41, 16], or a large number of samples to converge [64, 60, 41, 16]. These make them inappropriate or even infeasible for sampling in many applications.

Our developed algorithm only requires that it is possible to obtain the value of the function at each sampled point. No prior knowledge of the local or global levels of variations in the function is needed. This information is not available in many cases. In fact, the process of efficient distribution of sample points becomes apparent when this is available. Also, the only extra computational overhead required by this algorithm is the computation of a histogram at each stage of the sampling procedure. Finally, there is no acceptance or rejection step in the progressive sampling procedure, every obtained sam-

Table 2.1: Comparison of the Requirements for the Proposed ASHE Algorithm to other Adaptive Sampling Methods.

Requirements	ASHE algorithm	Entropy Based	Adaptive Farthest Point	Monte-Carlo methods	Gradient Based
Info. on Global fn.	*N*	*N*	*Y*	*Y*	*N*
Info. on Local fn. Variation	*N*	*N*	*Y*	*N*	*N*
Local/Regional Computation	*N*	*Y*	*N*	*N*	*Y*
Acceptance/Rejection Step	*N*	*N*	*N*	*Y*	*N*
Large No. Samples Needed	*N*	*Y*	*N*	*Y*	*Y*

Entropy based algorithms do further sampling based on local information content, while gradient based algorithms sample based on local gradients. The table entries represent N – Not required, and Y – Required.

ple is kept. This makes the procedure particularly useful for obtaining expensive samples. Table 2.1 shows a comparison of the ASHE sampling scheme to similar algorithms. The advantages of ASHE over the other algorithms are highlighted.

The described adaptive sampling algorithm is particularly useful in the following situations:

- No *a priori* information on the global, and relative levels of local variation of the function to be sampled is available.
- Obtaining samples is prohibitively expensive.

2.2 Theoretical Basis

Consider a monotonically increasing, non-linear function of $x \in \mathbb{R}^+$:

$$y = f(x) = Cx^n \,, \tag{2.1}$$

where C is a positive constant, and $n > 1$. An indication of the rate of change or level of complexity in the function is the derivative. This is given by:

$$f^{'}(x) = nCx^{n-1} \,. \tag{2.2}$$

Based on the assumptions made about the function,

$$x_i > x_j \Longrightarrow f^{'}(x_i) > f^{'}(x_i). \tag{2.3}$$

Hence, an adaptive sampling algorithm will attempt to place relatively more samples as x increases. If there is prior knowledge on the form of the function, an optimal sample distribution may be obtained based on these derivatives. However, without such prior knowledge, the obvious solution is to sample the function with even spaced samples in x. This results in the same sample density for the different regions. A more efficient scheme will result in a higher sample density as the values of x and $f^{'}(x)$ increase.

We propose an algorithm that increases the relative sampling density as the level of complexity increases. This algorithm is based on the distribution of the samples from the co-domain. Consider an example of the described function with, even space sampling in x, as shown in Figure 2.1(a), and even spaced sampling in the co-domain $f(x)$, shown in Figure 2.1(b). The functions are divided into three equal intervals in x. An equal number of 12 samples are used in both cases. Some of the samples are at the same locations

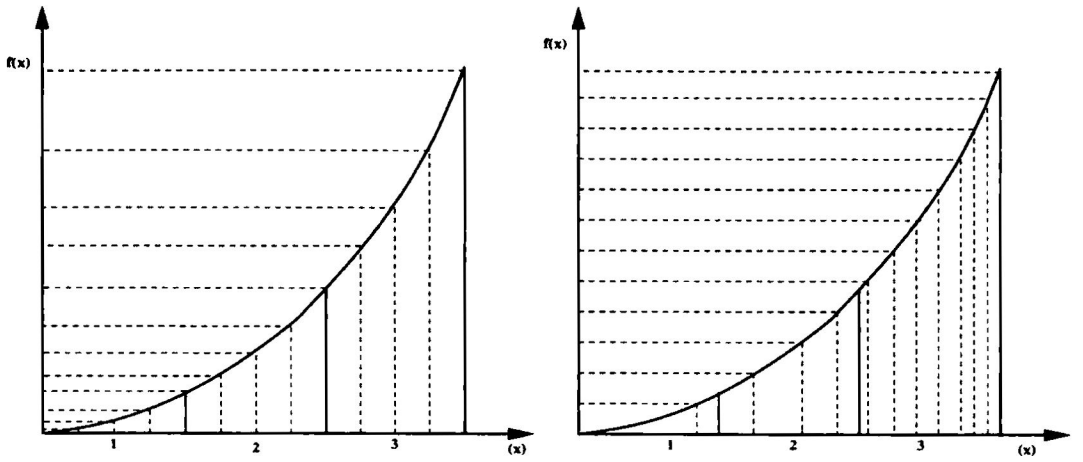


Figure 2.1: Illustration of basis for the ASHE algorithm in sampling a monotonic increasing function $x \in \mathbb{R}^+$ of the form $f(x) = Cx^n$, where C is a positive constant, and $n > 1$: (a) even spaced sampling in the domain, (b) ASHE algorithm, even-spaced sampling in the co-domain, resulting in more efficient sample density in the domain.

as the dividing red lines. As shown in Figure 2.1(a), even spaced sampling in x yields the obvious result of the same sample rate, even with increase in function complexity indicated by the increasing slope. That is, the three defined intervals, with different rates of change in sample values, have the same sample rate. Based on our discussions on error reduction, and adaptive sampling, this is inefficient.

Suppose we sample progressively, in order to equalize the distribution of the obtained samples. This will result in equal representation of samples in the co-domain. Hence, an equalized distribution of samples. This is shown in Figure 2.1(b). Projecting the even sampling in the co-domain $f(x)$ back to the domain x, shows a different sample rate for each of the regions. The relative increase in sampling density is proportional to the complexity in each region of the function, this is indicated by the slope. In contrast to other sampling schemes, the focus of the ASHE algorithm is on the co-domain, instead

of the domain.

In summary, the ASHE algorithm produces an adaptive sampling density in the domain, by varying the sample rates in *proportion* to the relative rate of change in the sampled function. This is an improvement over even spaced sampling, which produces the same sampling densities for regions containing different rates of change. However, the optimal *proportion* to determine the relative sample rate that minimizes errors will be specific to each function. Referring back to the derivative of the general form in (2.2), such optimal proportion will be a function of C, and n. We restate the fact that these are not known *a priori*. Also, any practical phenomenon to be sampled will consist of a complex combination of the type of function used in the illustration. A complete analytical consideration will have to consider these complex system. Note, however, that the extension of ASHE to such a complex system is valid. Samples are distributed in proportion to the relative levels of variation in the system. Finally, there is no prior knowledge of the exact divisions in the co-domain. This forms the basis of the histogram to be equalized. The foregoing precludes a rigorous mathematical consideration of the concept. We, however, conduct further analysis in a manner similar to that of other heuristic methods. An example of such analysis is found in [2]. These are usually performance and sensitivity analysis to determine factors that yield the best result from these algorithms, for a given class of applications. Many heuristics have been employed in solving practical problems for which obtaining an optimal solution is computationally prohibitive, or even infeasible. Some examples of such practical applications include routing for vehicles and in telecommunication networks [69, 11], and scheduling in industrial organizations [33]. Our analysis of the ASHE algorithm is reported in Chapter 4.

2.3 Illustration of ASHE

We illustrate the performance of the ASHE algorithm by comparing it to evenly spaced, and randomly spaced samples. The comparison is based on the quality of the functions that are reconstructed from sample points. Details of the actual implementation of the ASHE algorithm to generate these samples are presented later in Chapter 3. We experiment with two 2-dimensional functions, and reconstruct the function from their samples by the 4NN[1] nearest neighbor algorithm [43]. We emphasize the point that these examples are solely for the purpose of illustrating the algorithm, and not necessarily practical application areas. The areas of possible practical application will be discussed in the next section.

Figure 2.2 shows the comparison of the performance of ASHE to evenly spaced and randomly placed sample points. Note the clustering of the sample points in the regions of the functions with relatively higher local slopes when ASHE is used for sampling. Note also how the resulting normalized histograms compare to the Uniform distribution. Evenly spaced and randomly placed samples result in histograms with underrepresentation of the function values in regions of higher complexity, and a dominance of the function values in the regions of lower complexity. The histogram of the function values obtained by adaptive sampling however, shows a tendency towards the Uniform distribution. An objective measure of the histogram comparison is a sum-squared difference between the histograms and a normalized Uniform distribution with the same number of bins. The lower this value, the closer the histogram is to the Uniform distribution. The function reconstruction quality, indicated by the Peak Signal to Noise Ratio (PSNR) values, is highest when the function is sampled adaptively using the ASHE algorithm. A total of 100 experiments are conducted, and the indicated PSNR and deviation values are

[1] A sample point is reconstructed as the mean of the four nearest existing samples.

averages.

In our second experiment, we consider the image shown in Figure 2.3. This image represents a 2-dimensional function, whereby, the pixel grayscale values are the function values at each pixel location. A higher sampling rate is required in the parts of the image with dissimilar pixels because of the higher complexity. The image background is bland, and requires relatively fewer sample points. The image is of size $512 * 512 = 262,144$, 8 bits/pixels, and it is sampled at $16,384$ pixel locations indicating a ratio of $16 : 1$. The results of sampling adaptively based on ASHE are shown in the same figure. The performance is compared to the other sampling schemes as in the previous experiment. Note the efficient distribution of samples by the ASHE scheme as indicated by the cluster of sample points in the regions of high complexity - the face in the image. This has the required effect of a reconstructed image with better quality compared to the other sampling methods. Two numerical measures of image quality are used as a basis for comparison: the frequently used PSNR, and another measure of image quality called Structural Similarity (SSIM). This has been shown in [92] to be a better indicator of image quality than the PSNR. Both measures show that the image reconstructed from the adaptively sampled points based on the ASHE algorithm has the best quality. These numerical measures of image quality are supported by the better representation of the facial features in the image reconstructed from adaptively sampled points. The relationship between sample point positions and the resulting histogram of function values, in this case pixel grayscale values, is the same as those in the previous examples. That is, the histogram of the sampled grayscale values using adaptive sampling tends closer towards a Uniform distribution than the other sampling methods. A total of 100 experiments are conducted. The indicated PSNR, SSIM, and deviation values are averages of these experiments.

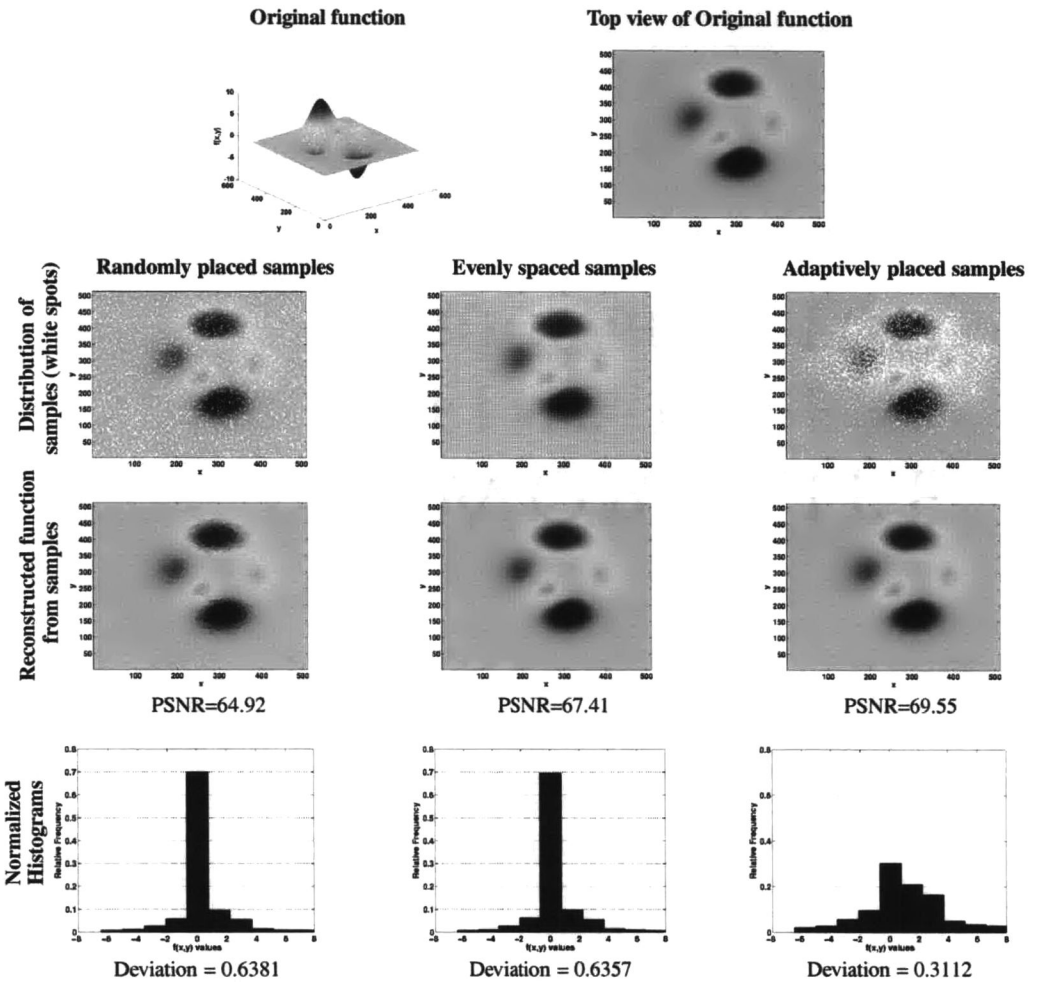


Figure 2.2: Performance comparison of the adaptive sampling algorithm to randomly placed, and evenly spaced samples for sampling a 2-dimensional function with varying slope. There are 512 values for both x, and y, yielding 262144 values. The function is sampled at 4096 locations, and reconstructed using the nearest neighbor method. Hereby, the Peak Signal to Noise Ratio (PSNR) is chosen as the objective measure of the quality of the reconstructed function. The Deviation values are defined as the mean squared deviation of the histograms from a normalized uniform distribution with the same number of histogram bins.

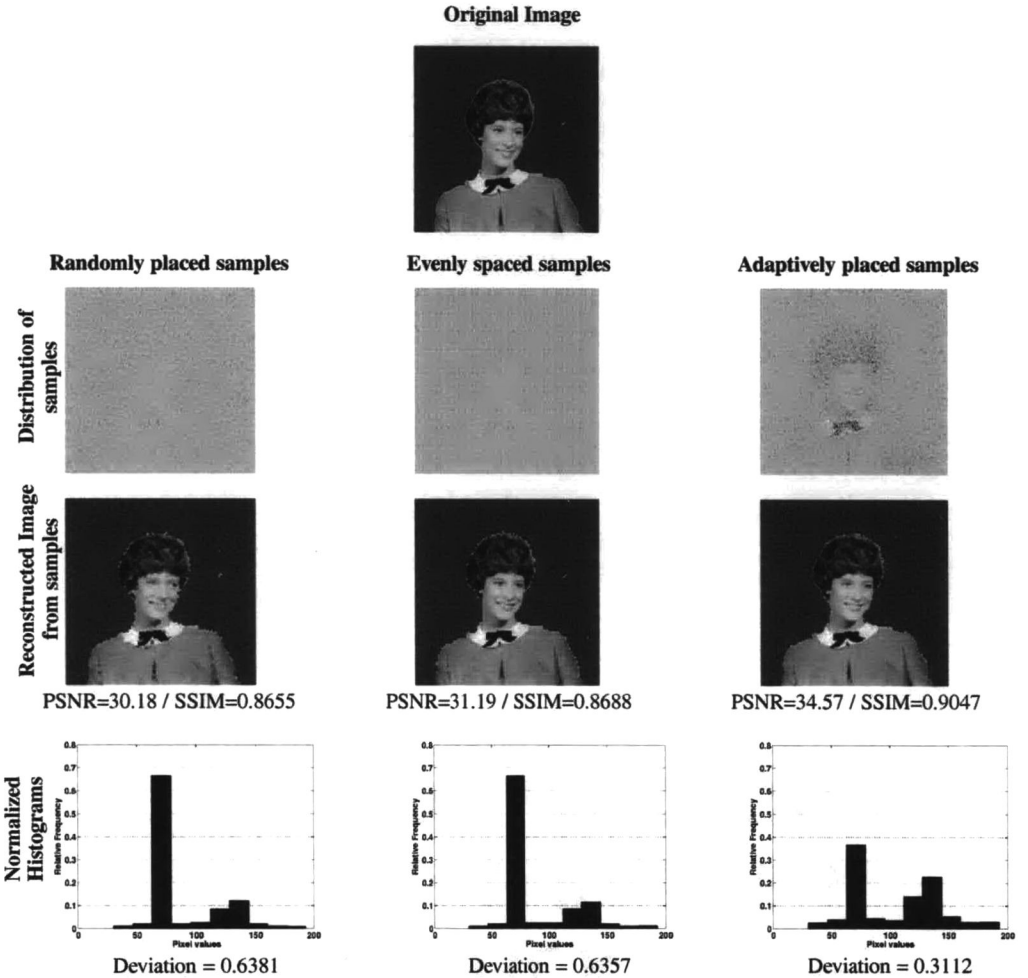


Figure 2.3: Performance comparison of the ASHE algorithm to randomly placed and, evenly spaced samples for sampling a 2-dimensional grayscale image. The original $512 * 512 = 262144$ pixels are sampled in 16384 locations, a ratio of $16 : 1$. The images are reconstructed using the nearest neighbor method. PSNR, and structural similarity (SSIM) are objective measures of the reconstructed image quality. The Deviation values are the mean squared deviation of the histograms from a normalized Uniform distribution with the same number of histogram bins.

2.4 Application Areas

As stated earlier, the examples in the previous section only serve as illustrations of the concept, and are not practical application areas. Based on the advantages of the ASHE algorithm stated in Section 2.1, we have identified some practical areas of applications. The following discussions are fairly generic, it should be straightforward to adapt the algorithm to specific problems.

2.4.1 Data Synthesis

Many forms of data analysis require adequate, and statistically representative population of the dataset in question to be able to make reliable inferences, and draw general conclusions. The problem in many fields of study is that the amount of available data does not fulfill the these stated requirements. Cost, time, and other limitations on resources may be prohibitive to the collection of the data. This problem has been solved in many instances by generating synthesized data. The requirement of statistical representation is usually that of maximum diversity in the dataset i.e., an equal representation of all possible members of a population. Maximum diversity is required in sample data in order to ensure that results from such are representative of the entire domain. Analysis of such data can then lead to inferences and conclusions that take all possible output scenarios into account. As demonstrated in Section 2.2, sampling to maximize diversity i.e., equalized distribution of obtained samples results in more efficient sampling. An image synthesis application that employs the ASHE algorithm is the subject of Chapters 5 - 7 of this dissertation.

2.4.2 Design of Experiments

Experimental results are usually functions of various factors. For example, a chemical reaction or biological process may depend on such factors as temperature, pressure, presence of catalyst or other reagents. It is usually required to determine the results of such experiments over a range of factors. It may be expensive or impractical to perform these experiments over all possible ranges and combinations of these factors. The results of such experiments or processes can be modeled as a multi-dimensional function with each dimension being one of the factors. Usually, there is no *a priori* information on the global, and relative levels of local variation of the outputs from these experiments. Regions of change due to a factor or combination of factors are usually of interest in these experiments. With the constraint on the number of experiments, it will be beneficial to perform more experiments in regions of this multi-dimensional space where there is relatively more change in the experimental results. This space can then be progressively sampled using the ASHE algorithm, whereby each subsequent sample location, that is, combination of factors for which the experiment is performed is determined by the current distribution of already obtained samples.

2.4.3 Surface Reconstruction from Expensive Samples

Computational Physics, Chemistry and Biology involve studies that require the computation of surfaces representing various phenomena. An example of this is the computation of the potential surface of a molecule in a particular electron state using first principles, that is, ab initio computations. These surfaces are usually multi-dimensional, and are constructed from the computed phenomenon values at various sample points in the space. In many cases, obtaining these values at each sampled point is computationally expensive. Also, no *a priori* information on the global or local variation on this surface is

available, only the ability to compute the surfaces' value at each sampled point. To minimize the surface reconstruction error from the points at which the surface values have been calculated, or to adequately represent regions of transition, it is required to compute relatively more values in regions where there is more change in the surface values. The ASHE algorithm can be used to determine the sample points where the surface values are to be computed by ensuring efficient variable sample rates.

2.4.4 Progressive Transmission/Rendering

Transmission of image data on a limited bandwidth channel can be effectively achieved by progressively sampling the image using the ASHE algorithm. This results in the more important information from the image being transmitted earlier. With this approach, truncating the data stream will only lead to the loss of the less important part of the information stream needed for reconstruction. The approach yields similar results in image rendering, with the more important region being rendered earlier such that a profile of the image is quickly represented. Similar work has been done with a different approach in [42].

CHAPTER 3

MODELS UTILIZED IN IMPLEMENTING ASHE

In this chapter, we discuss the three models employed in implementing the function values equalization described in the Adaptive Sampling by Histogram Equalization (ASHE) algorithm. All the described models are generally utilized in many areas of optimization, especially for problems in which directly obtaining optimal solutions is infeasible due to the computational cost. For these problems, the models are used to obtain variables that minimize or maximize a function. We apply variants of these models to obtain an efficient sample distribution as described in the ASHE algorithm. Though somewhat different, our problem may also be seen as an optimization problem, in which we intend to maximize the efficiency of the sample distribution. This efficiency is defined based on an objective measure. The set of sample points obtained by these algorithms constitutes a set of solutions.

The underlying concept that each model is based on is presented. Any variations or modification of the general form for our specific purpose is stated, and justified. General examples are presented to aid in the understanding of these models. The specific details of implementing the ASHE algorithm with each model are then described. Examples to illustrate the ASHE implementation with each model are, however, delayed until the discussion on their performance and sensitivity analysis in Chapter 4.

3.1 Active Walker Model

The active walker model can be explained both in the framework of the motion of Brownian particles, and as a variant of the random walk [72]. The simple Brownian motion

will not result in any structure required to model the systems to be studied. Therefore, Brownian particles "*with the ability to generate self-consistent fields, which in turn influence their subsequent movement, physical and chemical behaviors*" are introduced [75]. These are called *active Brownian particles*. The term *active walker* was first introduced in this work [29], in which a discrete approximation of the motion of these active particles was used to model a complex system. The active walker model has been used to simulate, and analyze numerous complex systems in both the physical and life sciences [48, 49, 46, 79, 78, 37, 38].

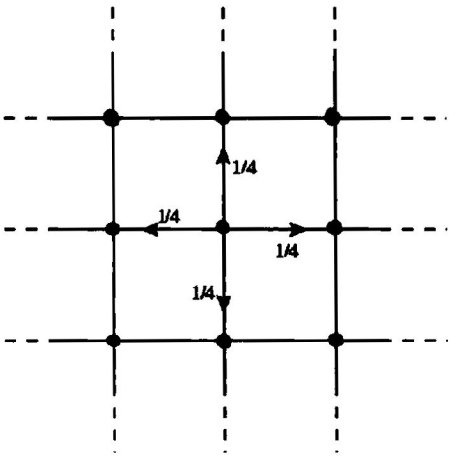


Figure 3.1: Simple symmetric random walk on $\mathbb{Z}^2$

The random walk approach is, however, the more appropriate of the two frameworks to explain our specific use of the active walker model. Consider the simple, symmetric random walk on $\mathbb{Z}^2$ shown in Figure 3.1 [72]. From the starting central position, the random walk can be seen as a specific case of a Markov-chain [63], in which the transition probabilities are given as follows:

$$p_{ij} = \begin{cases} \frac{1}{4} & \text{if } \|i-j\| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Each step taken in a random walk is discrete, and of equal sizes. The chain starts with an equal probability of moving into any of the vacant positions. The generalization of this to a higher-dimensional space is trivial. In their most general form, active walkers are pseudo-random walkers with the following properties:

- They take discrete, but not necessarily equal step sizes.
- The direction of their movement may be either random or biased.
- Their step sizes and direction of movement may depend either on local information contained in their current location, or global information in the walking space. They may also depend on a combination of these.
- In the case of multiple active walkers, the behavior of each walker may depend on peer interaction.
- Multiple active walkers can not occupy the same location at the same time.

All the variable properties, such as the movement, are governed by defined fitness criteria. For example, in a situation where moving charged particles are simulated, the distance and direction moved by an active particle may depend on the charge carried by the particle and those in its vicinity [79]. The result is a pseudo-random walk, which is biased based on the fitness criteria. The adaptive or biased random walk approach has been used in solving optimization problems [12].

3.1.1 ASHE Implementation using Active Walkers

For our specific application, simulated active walkers are employed to implement the ASHE algorithm by placing sample points in the location of the walker in the sample space. Initial samples are obtained by placing even spaced active walkers in the space.

A histogram is formed from the function values obtained from these initial locations. We establish a fitness criterion based on the state of the histogram. Whereby, after each sample addition, the normalized histogram of samples is updated, and compared to a normalized Uniform distribution with the same number of bins. The comparison of the two histograms results in a fitness criterion FC which is given by:

$$FC = \sqrt{\sum_{i=1}^{n} (\hat{h} - h_i)^2}\ , \tag{3.2}$$

where n is the number of bins in the histograms, h_i are the relative frequencies from the sample distribution, and $\hat{h} = \frac{1}{n}$ is one of the equal valued relative frequencies from the Uniform distribution. FC has a lower bound of zero. A decrease in the value of the FC indicates that the newly added sample moved the histogram of samples closer to the normalized Uniform distribution. The active walker that obtained the sample then moves a *short* step in order to sample more in its current vicinity. An increase in the value of the FC due to the addition of a sample, indicates a deviation of the distribution of samples from the Uniform distribution. The active walker that obtains the sample resulting in the deviation is made to sample in a location different from its current vicinity by taking a *long* step. The definitions of the terms "short", and "long" steps will be addressed in detail in Chapter 4, while discussing the sensitivity analysis. The distance between the current and subsequent locations of a walker is determined as the resultant of vector lengths along each dimension. The direction of each vector is randomly chosen, independently, resulting in a random direction for the resultant vector. This process is then continued to progressively sample the space until the required number of samples is obtained. Multiple active walkers are usually employed to ensure that the entire sample space is covered. A variant of the self-avoidance mechanism is also included to

ensure that a location is not sampled multiple times [72]. In summary, the active workers employed to implement ASHE have the following specific behaviors:

- The position of an active walker represents a sampled location.
- There is no cost associated with the distance moved by an active walker.
- A self-avoidance mechanism is implemented to ensure that a location is not sampled multiple times.
- The next location of an active walker is dependent on its current location, and the step size adapting criterion. The adapting or fitness criterion is the change in state of the distribution of function values.
- Their direction of movement is random.

Algorithm 1 shows the implementation of ASHE using the active walker model.

Algorithm 1 . Active walker model implementation of the ASHE algorithm

Initial definitions:

Objective function, e.g. function value
Variables/factors the objective function is dependent on
Range and possible values that all variables/factors can take

Sampling initialization:

Obtain initial randomly located samples using active walkers
Compute/obtain the objective function values from initial sample points
Compute normalized histogram from initial sampled function values
Compute Overall Fitness Criterion OFC

while Sample points $\leq$ required no. of samples **do**
for all Active walkers **do**
Obtain new sample point
if Location has already been sampled
Obtain closest unsampled location
(random choice if multiple unsampled locations exist at same distance)
end if
Add new sample from active walker to existing samples
Compute new normalized histogram after single addition, and
Compute New Fitness Criterion NFC
if $NFC < OFC$
Single walker takes *short* step size in random direction
else Single walker takes *long* step size in random direction
end if
end for
Compute new overall normalized histogram
Compute OFC
end while

3.2 Ant Model

This model is based on the behavior of ants in search of resources, usually food. Many insect species deposit a substance called pheromones when walking to or from food sources [40, 32]. The role played by this mechanism on their ability to efficiently search for food has been studied [31, 14]. Many of these insects, for example ants, possess little or no sense of sight, and communicate primarily through their sense of smell. Consider the simple case of an ant colony and a resource, say food, as shown in Figure 3.2. The ants can reach the food by either of the two paths, with one being much longer than the other. The ants deposit pheromones as they traverse these paths. Assume an initial random access of the paths, and also assume that the effect of deposited pheromone spreads, and fades with time, due to a diffusion process. The pheromone update over time, is generally modeled as:

$$\tau_{ij} \longleftarrow (1-p).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \,, \tag{3.3}$$

where i and j are endpoints of the path, p is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^{k}$ is the quantity of pheromone laid on the path (i,j) by ant k. It is expected that the pheromone concentration on the shorter path will be refreshed more often, therefore, the maintained concentration level will be higher. This will attract more of the ants into using this path to get to the food. Hence, the optimal route to the resource is established. Probabilistic models of these kind of behavior have been developed, and they show that the initial equal probability of taking either path is updated to increase the probability of the shorter path [31]. In summary, the colony of ants communicate indirectly to reinforce a good solution by modifying their environment through positive feedback.

This kind of communication is referred to as *stigmergy* [39]. When compared to other

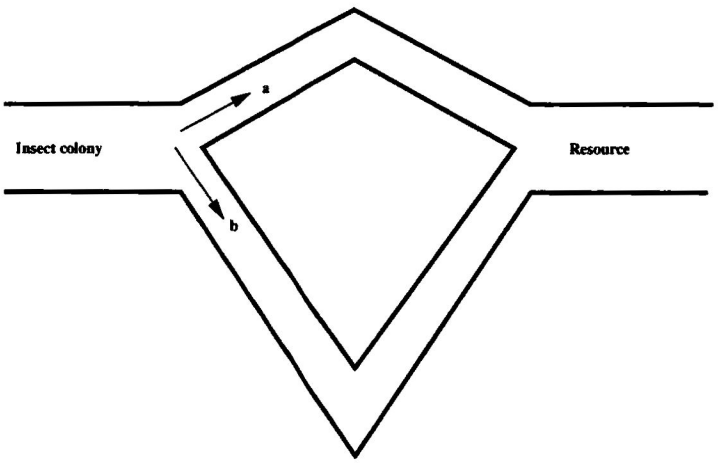


Figure 3.2: Illustration of 2 possible paths for ants to reach a resource. With path 'a' shorter than 'b'

forms of communications, it is noted to have two unique characteristics [17]. These are:

- It is indirect, and non-symbolic, insects communicate by modifying their environment.
- It is local, that is, the information can only be accessed by insects that visit the vicinity of the pheromone *footprint*.

Algorithms based on the ant model have been successfully used in solving numerous optimization problems [19, 86, 55]. These fall under the generic name of Ant Colony Optimization (ACO) algorithms. Some variants of this are the Ant System (AS) [20], and the Ant Colony System (ACS) [18]. A good review of the progress in this area of study can be found in [17]. A basic assumption made by all these algorithms is that the ants live in an environment where time is discrete.

3.2.1 ASHE Implementation using the Ant Model

In implementing the ASHE algorithm, the described 1-dimensional path model is extended to multiple dimensions. Whereby, ants *forage* in the space to be sampled, and samples are obtained from their current locations. At the start of the sampling process, each sample location is allocated equal probability of being foraged. The probabilities are modeled as pheromone concentrations. The sampling is done in discrete time intervals. The *resource* is the obtained sample. Whether a positive feedback is sent by an ant depends on the change in the state of the distribution of the already obtained samples. The same fitness criterion FC defined in (3.2) for the active walker model is used. In this case, a positive feedback is only generated if the obtained sample moves the updated distribution closer to the Uniform distribution. The ant modifies its environment by depositing pheromones in the multidimensional vicinity of where it obtains a *good* sample, indicated by a reduction in the value of FC. The effect of this deposit is a relative increase in the probabilities associated with the sample locations in the vicinity of this sample. The questions concerning the amount of increase, and the extent of the spread will be addressed in the performance and sensitivity considerations in Chapter 4. The increase is highest in the locations nearest to where the sample was obtained, and tapers down at a non-linear rate. All locations that are already sampled are allocated a zero probability to avoid multiple sampling of the same location. At each discrete time step, the probabilities associated with the unsampled locations are reduced by the same factor. This is to simulate the process of the pheromone evaporation with each time step. Subsequently, the ant samples in its vicinity in a manner that reflects the probabilities associated with the sample locations. That is, where there are multiple non-zero probability locations of the same distance, it samples the location with the highest probability. Otherwise, it samples in the only non-zero probability location in its vicinity. The defini-

tion of vicinity will be clarified in the discussions on sensitivity analysis. It is expected, that the effect of the feedback created by an ant generally goes farther than its movement in any one time step. The ant model implementation of the ASHE algorithm is shown in Algorithm 2.

Algorithm 2 . Ant model implementation of the ASHE algorithm

Initial definitions:

Objective function, e.g. function value
Variables/factors the objective function is dependent on
Range and possible values that all variables/factors can take
Associate equal probabilities with all sample locations

Sampling initialization:

Obtain initial samples from location of randomly placed ants
Compute/obtain the objective function values from initial sample points
Compute normalized histogram from initial sampled function values
Compute Overall Fitness Criterion OFC

while Sample points $\leq$ required no. of samples **do**
for all ants **do**
Obtain new sample point from non-zero probability location in *vicinity*
if Multiple non-zero locations exists
Obtain sample from location with highest associated probability
end if
Add new sample from ant to existing samples
Set probability associated with sampled location to zero
Compute new normalized histogram after single addition, and
Compute New Fitness Criterion NFC
if $NFC < OFC$
Increase probabilities associated with locations around sample
by values that sum up to 1 in the surrounding sample locations
end if
end for
Multiply probabilities associated with sample locations by constant < 1
Normalize probabilities based on non-zero locations in the space
Compute new overall normalized histogram
Compute OFC
end while

3.3 Evolutionary Algorithm Model

The class of evolutionary algorithms (EAs) includes genetic algorithms [61], genetic programming [47], evolutionary strategies [30], and evolutionary programming [28]. Common to all the variants of this class of algorithms are elements of the principles of natural biological evolution. They operate on populations based on the principles of survival of the fittest. Population members deemed to be best suited for surviving in a particular environment form the basis for creating the next generation. Evolutionary algorithms model this natural process by applying principles like recombination, mutation, and migration. A good introduction to evolutionary algorithms [3] addresses these basic concepts.

EAs have been used variously to solve search and optimization problems [59, 84]. In general, they consider a population of possible solutions, and remove the poor solutions based on some fitness criterion. The surviving population members then form the basis for producing a new generation. The new generation is produced primarily by combining surviving members. The rationale for this is that combining elements from fit members will result in even fitter members. A mutation process is also used to generate new members. This is an occasional perturbation that results in a new member, whose properties are not completely accounted for by any existing member. The population size may be kept constant or varied over the generations. This death, survival, and mutation process continues until an acceptable solution is obtained.

3.3.1 ASHE Implementation using the Evolutionary Algorithm Model

In implementing the ASHE algorithm using an evolutionary algorithm approach, we start the sampling process in even spaced locations. The coordinates of these locations serve

as the population in the first generation. Again, the fitness criterion is the same as in the other models, and stated in (3.2). That is, the change in the FC value due to the addition of a sample from a location determines whether it is considered fit, or not. Based on the prior discussions, a sample from a location that results in the reduction of the FC value is considered fit. The recombination process used in the standard genetic algorithms will not necessarily produce a new fit member. To illustrate this point, consider two fit members of a current generation. For simplicity, let us assume an even number of dimensions in the space to be sampled, say two. The population members consist of an ordered pair of integers. Recombination between two fit members will yield new offspring that do not necessarily have any relation to the parents. For our purpose, recombination between two fit members may yield new locations that have no bearing on the original location. This is shown in Figure 3.3.

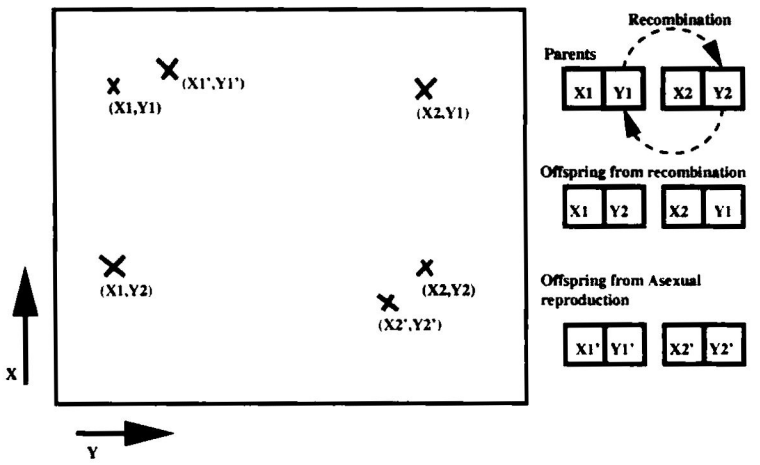


Figure 3.3: Illustration of standard genetic algorithm recombination process, and an Asexual process better suited for implementing ASHE.

A different approach of producing a new generation has to be taken. This is so because the ASHE algorithm requires that further sampling is done in the neighborhood of a current good solution as indicated by the fitness criterion. We take the approach of

making the fit population member reproduce in an asexual manner. That is, a fit member produces offspring in its current vicinity. The number of offspring, and how far their neighborhood can be from their parents, are variables that will be discussed under the performance and sensitivity analysis. A very small probability that the offspring of a fit parent may die is also introduced. All unfit parents die without offspring, but the sample from their current location is accepted. A random selection of the starting population size is made from the offspring to ensure that the population size remains the same for every generation. Algorithm 3 shows the evolutionary algorithm model implementation of ASHE.

Algorithm 3 . Evolutionary algorithm model implementation of ASHE

Initial definitions:
Objective function, e.g. function value
Variables/factors the objective function is dependent on
Range and possible values that all variables/factors can take

Sampling initialization:
Obtain initial randomly located samples/starting population
Compute/obtain the objective function values from initial sample points
Compute normalized histogram from initial sampled function values
Compute Overall Fitness Criterion OFC

while Sample points $\leq$ required no. of samples **do**
for all Members in current generation **do**
Obtain new sample point
if Location has already been sampled
Obtain alternate, close sample point
end if
Add new sample from population member to existing samples
Compute new normalized histogram after single addition, and
Compute New Fitness Criterion NFC
if $NFC < OFC$
Parent reproduces/divides (Asexual reproduction) in its vicinity
Generate random number S between 0 and 1
if S > prob. of survival (>= 0.7)
Offspring dies
end if
else Parent dies
end if
end for
Make random selection of starting population size from offspring
Compute new overall normalized histogram
Compute OFC
end while

CHAPTER 4

PERFORMANCE AND SENSITIVITY ANALYSIS OF MODELS

Here, we establish two measures of the performance of the models in the efficient distribution of sample points. In contrast to the basis of comparison in Section 2.3, these are independent of any reconstruction algorithm. The measures here are based on the entropy measure of information [34, 82], and the Nyquist-Shannon minimum sampling rate for band-limited signals [67, 56]. These serve as an indication of the relative levels of variation in a sampled function. For both measures, a high value will signify more complexity, thus requiring a relatively higher sample density. The purpose of the ASHE algorithm is to efficiently distribute sample points. That is, adapt the sample density such that they reflect the local, and global levels of variation in the space being sampled. The defined objective measure of performance is thus based on the correlation between these indicators of variation and the sample density. A high correlation between the sample density and either of the two measures will signify good performance of the sampling scheme, and vice-versa.

Based on the established performance criteria, we carry out a performance and sensitivity analysis of the models. The sensitivity analysis seeks to investigate how such performance varies with change in factors in the different models used in the ASHE implementation. The results from these are presented, and conclusions are drawn where appropriate.

4.1 Measure Based on Frequency Content

The Fourier transform (FT) is used to decompose a function in time or space into its sine and cosine components of different frequencies. The space or time varying function can then be represented in terms of its frequency components. This is called a frequency domain or spectral representation of the function. We use 2-dimensional functions, specifically 2-dimensional digital images to illustrate the use of this transformation in establishing a measure of complexity. Since we are considering digital images, we will further restrict the discussions to the Discrete Fourier Transform (DFT), which is a sampled version of the continuous FT. Consider an image of size $M \times N$, then the DFT is given by

$$F(k,l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-2\pi i(\frac{mk}{M} + \frac{nl}{N})}, \tag{4.1}$$

where $f(m,n)$ is the image in the spatial domain, and the exponential term multiplying it is the basis function corresponding to each $F(k,l)$ in the frequency domain. The basis functions are sine and cosine waves with increasing frequencies from $F(0,0)$, which is the DC component[1], to a maximum of $F(M-1, N-1)$. The DC component represents the average brightness in the image. The resulting Fourier transform is complex, containing the real or magnitude, and imaginary or phase components. The size of each component is the same as that of the original image.

Figure 4.1 shows two images and the corresponding DFT magnitude images. For simplicity, the images contain single spatial frequencies. The original DFT magnitude images have the DC components at the edges and the highest frequency components in the middle. These have been shifted using the MATLAB fftshift function so that the DC

[1]The zero frequency component of a signal is also referred to as the Direct Current - DC component.

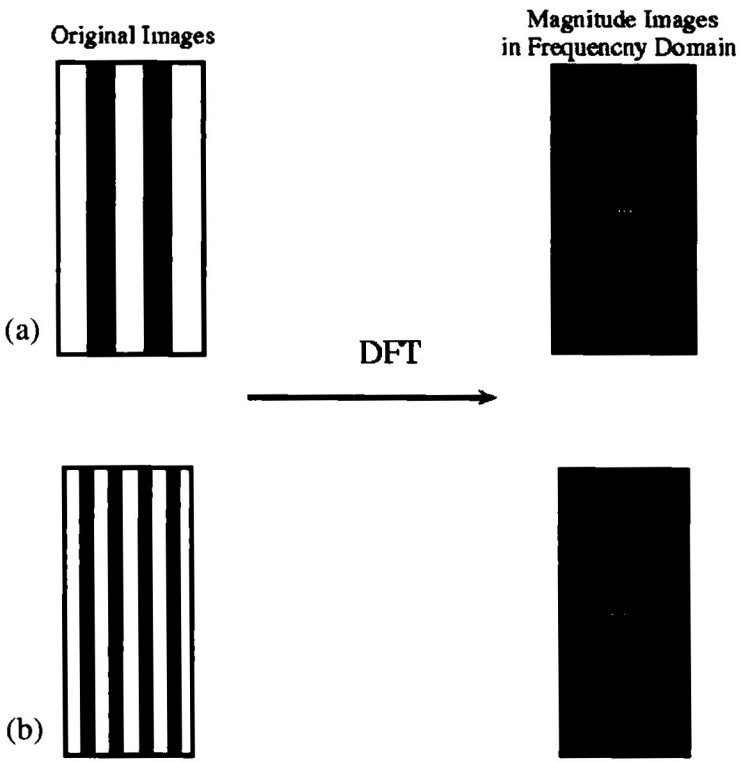


Figure 4.1: Shows two images of single spatial frequencies, and the corresponding magnitude images in the frequency domain. The image in (a) has a lower frequency than that in (b), this is reflected in the distance of the fundamental frequencies from the DC component.

components are in the middle, and the highest frequency components are at the edges. The pixels in the middle of the DFT images are the brightest, indicating that the images are dominated by their DC components. Note the two bright spots on either side of the DC component. These represent the fundamentals of the single spatial frequencies, and are mirror images of each other. Their distances from the DC component is an indication of the frequency they represent, the higher the spatial frequency, the bigger this distance. This explains why the distance is larger in the DFT from the image with the higher spatial frequency. To aid in the general interpretation of DFT images, consider the cross-sections of the power image of 2-dimensional DFTs as shown in Figure 4.2. Figure 4.2(a) gives a simplistic representation of a single frequency image in the frequency domain. Figure 4.2(b) shows multiple power plots, each representing multiple frequencies.

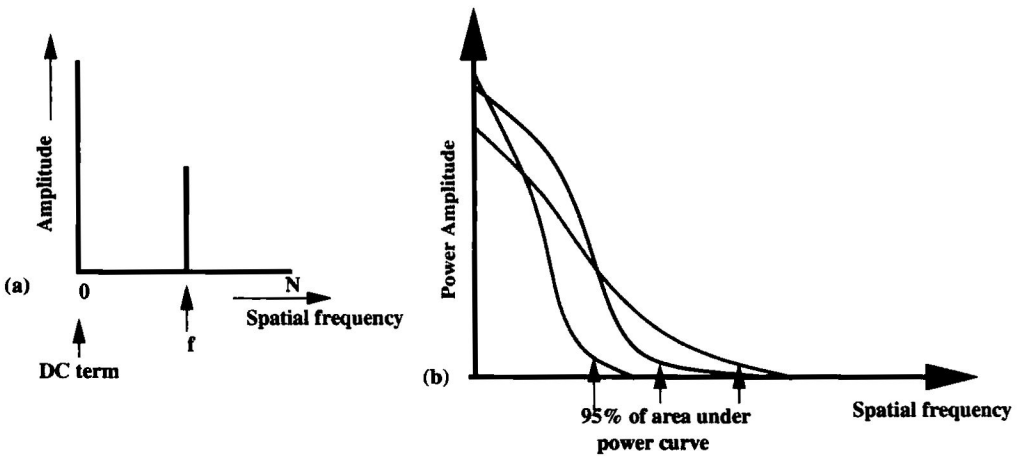


Figure 4.2: Cross-sections of images in frequency domain:(a) simplified frequency representation, (b) multiple plots showing different power spread in their spectrum

Our metric is based on the power distribution in the frequency components of the functions in question. We determine this by summing up the power in the spectrum, starting from the lowest to the higher frequencies until a value of 95% of the total power in the spectrum is obtained. We do not use 100% of the power since the function may have

support in the entire frequency domain. The DC component is also excluded to avoid a bias due to differences in function amplitudes. Generally, the higher the distance, that is spatial frequency at which this value is attained, the more the high frequency components in the function. We divide the function into 16 regions of equal sizes, and compute this value for each region. Our objective measure is the correlation coefficient (CC) between the relative sample densities in the regions and the values indicating the frequency content. The correlation coefficient takes values ($-1 \leq CC \leq 1$). A positive CC value indicates that the sample densities are higher in the regions where the function contains high frequency components. The higher the positive correlation value, the more efficient the sample distribution obtained by the employed model.

A drawback of using the DFT approach is the classic time/spatial versus frequency resolution trade-off. That is, computing the Fourier transform over a small time/spatial window will result in poor frequency resolution but good time resolution. Increasing the window size improves the frequency resolution at the expense of the time resolution. The frequency resolution is more important for the application. The window size we can use is constrained because the function has to be sub-divided into regions. We apply some zero-padding in the spatial domain before the DFT transformation. The zero-padding does not improve the frequency resolution, but it does increase the sampling rate, leading to appreciable improvements in our results.

Figure 4.3 shows two of out test functions, an image, and a 2-dimensional energy function. The functions are divided into 16 regions for the purpose of obtaining the frequency based objective measure by region. Also shown are the corresponding complexity measure images, in which the shades indicate the level of complexity. Note the general visual correlation between the frequency-content based measure, and the apparent regions of complexity in the functions.

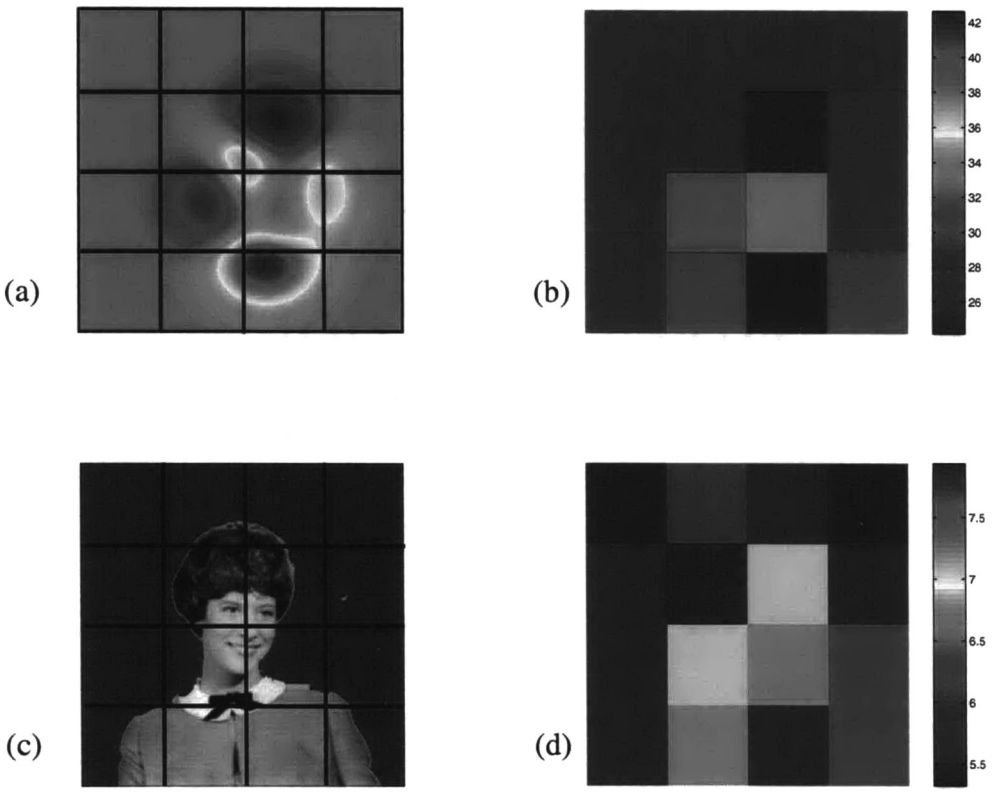


Figure 4.3: Functions divided into 16 equal regions (a) and (c), the corresponding images based on frequency content associated with the regions (b) and (d) respectively.

4.2 Measure Based on Entropy Measure of Information

Entropy as a measure of information content in a discrete system is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{4.2}$$

where X is a discrete random variable that can take possible values $x_1, x_2, ..., x_n$, and $p(x_i)$ is the probability that X takes the value x_i. The concept may be understood intuitively in terms of uncertainty. If the outcomes $x_1, x_2, ..., x_n$ are equally probable then uncertainty is high, and the entropy is maximal. If, however, an outcome is certain, the entropy is zero, which indicates that no additional information is obtained from the outcome.

For our purpose the function in question is divided into 16 equal sized regions as described for the frequency based measure. The entropy of each region was computed. Higher entropy values indicate greater uncertainty or complexity in the functions. An objective measure was obtained by computing the correlation coefficient between the entropy values and the relative sample densities. Again, higher positive correlation values indicate a more efficient sample distribution. Figure 4.4 shows the example functions and the corresponding entropy images. The entropies associated with each region are represented by the color shades. Note the visual correlation between the region of complexities in the functions, and their corresponding entropy shades.

4.3 Analysis of the Active Walker Model

We experiment with two functions: a smooth energy function, size 512×512 pixels, and a 2-dimensional image of size 256×256 pixels, to represent rapidly varying functions. We obtain a total of $4,096$ sample in each case, representing a ratio of $1 : 64$, and

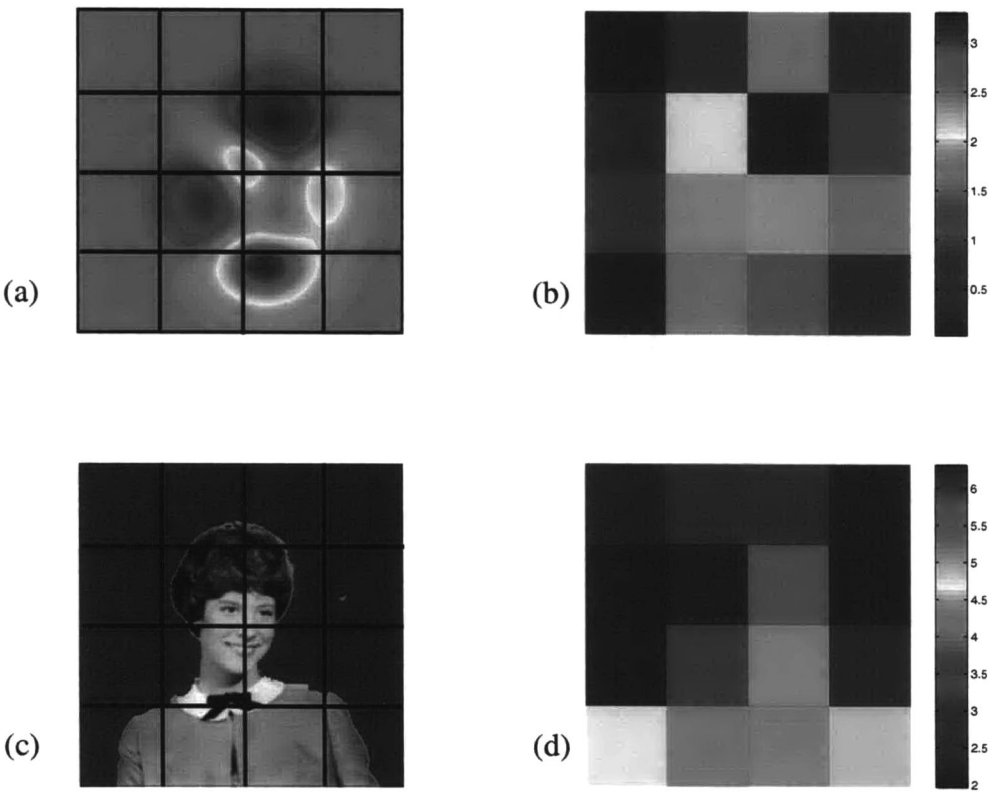


Figure 4.4: Functions divided into 16 equal regions (a) and (c), the corresponding images based on entropy associated with the regions (b) and (d) respectively.

1 : 16 respectively. The two test functions are shown in Figure 4.5. We emphasize that these test functions are used for the purpose of experimentation alone. The key elements expected in the areas of practical application are not present in them. That is, obtaining the sample values is not expensive, and we have complete prior information on these functions. They however, fully meet the requirement for these experimental purpose.

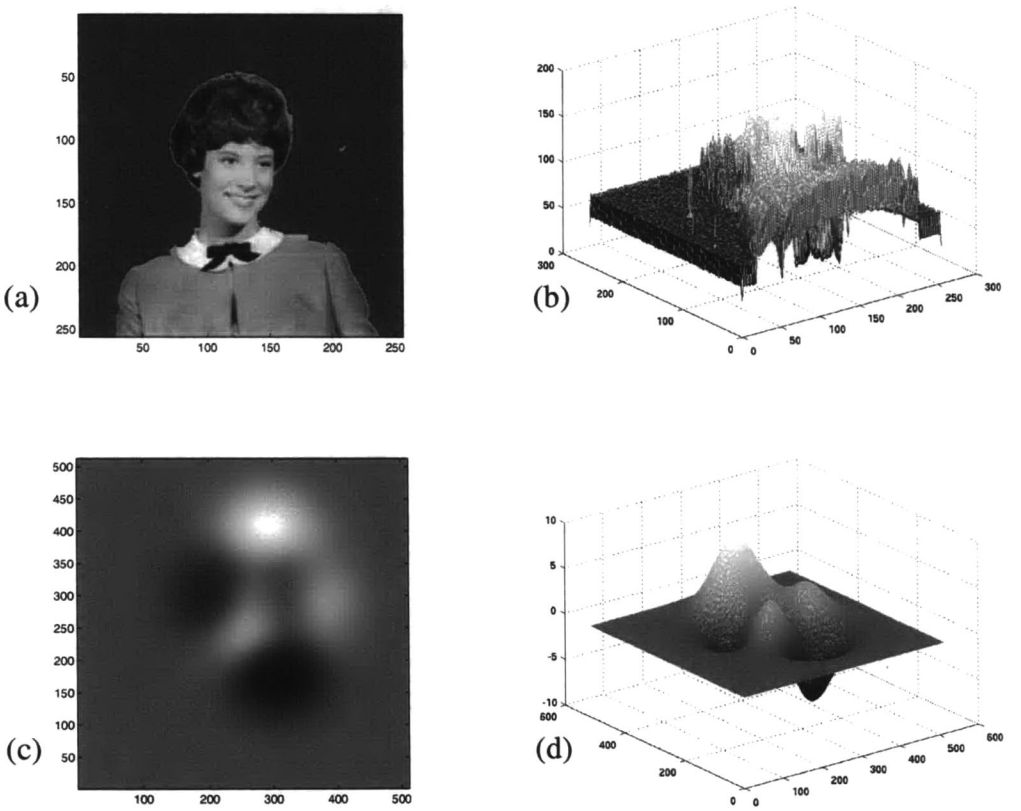


Figure 4.5: Test functions, top views (a) smooth function, (c) rapidly varying function, and their corresponding side views (b) and (d) respectively.

We identify four key factors that may affect the performance of the active walker model as described in Section 3.1.1. These are:

1. Number of active walkers (N_{aw}). This will determine the number of steps taken in the adaptive sampling process. Starting with n walkers means that $(4,096/n) - 1$ steps will be taken.

2. Number of bins (N_{ub}) in the histogram to be equalized. This is important because it constitutes a form of *resolution.* It is the number of unique groups in which the sample function values are divided.

3. Large step size (LSP). This is the step taken away from a vicinity due to an increase in the fitness criterion value. We define this as a function of the size of the space being sampled.

4. Small step size (SSP). This is the step taken in order to sample in the current vicinity of an active walker, due to a decrease in FC, signifying an improvement. This is also defined as a function of the size of the space being sampled.

We experiment with the following values of these factors:

$N_{aw} = \{4,\ 64,\ 100,\ 144,\ 1,024\}$

$N_{ub} = \{2,\ 8,\ 16,\ 32,\ 256\}$

$LSP = \{0.2,\ 0.25,\ 0.3,\ 0.35,\ 0.4\}$

$SSP = \{0.02,\ 0.04,\ 0.06,\ 0.08,\ 0.1\}$

Both LSP, and SSP are obtained by multiplying the vector containing the sizes of the dimensions of the space with these numbers. For each experiment, the function is sampled using the active walker model with a combination of these factors. One complete set of experiments thus includes 625 runs. In order to ensure adequate statistical representation, we run 100 complete sets, that is 62,500 runs in all. Figures 4.6 and 4.7 show examples of the progression of the adaptive sampling for both test functions. In both examples, we used the following values in the active walker model: N_{aw}= 64, N_{ub}= 16, and LSP= 0.4. We used SSP= 0.06 for the smooth function, and SSP= 0.02 for the image. Note that nine images from intermediate stages are shown instead of the total of 64 expected in the sampling process with N_{aw}= 64.

In both examples, the sample density changes to reflect the complexity in the sampled function as the process progresses.

We sort the results of the tests by the measures based on frequency content, and entropy. These results are shown in Tables 4.1 - 4.4. The tables show the results for the two test functions, sorted by the two measures of performance, hence the four tables. Due to space constraint, we only show the top and bottom 20 runs based on the sorting criteria. The important information to aid in our analysis is, however, all contained in the shown portion of the results.

Note that the correlation between the sample density and the frequency based measure is lower than that between the sample density and the entropy based measure. This is probably due to the limitations of the frequency based measure for this particular purpose, as stated in Section 4.1. The correlation between both measures is generally good as the sorted table indicates. That is, both objective measures essentially give the same information.

Tables 4.1 and 4.2 show the results for the experiments using the smooth function. The results show that the active walker model requires more than three steps to achieve a good solution. This is indicated by the number of appearances of $N_{\mathrm{aw}} = 1,024$ at the bottom of the table. $1,024$ active walkers would only take $(4,096/1,024) - 1 = 3$ steps each to complete the sampling. The sample density distribution is still essentially random after three steps. Generally, more steps improve the ability of the adaptive process in efficiently distributing samples. This has to be balanced by the need to cover the sampling space, as too few active walkers may get stuck in a locality. The results show that for this application, at least about 25 steps will yield a good sample distribution. The number of active walkers should be chosen relative to the total number of samples to be obtained.

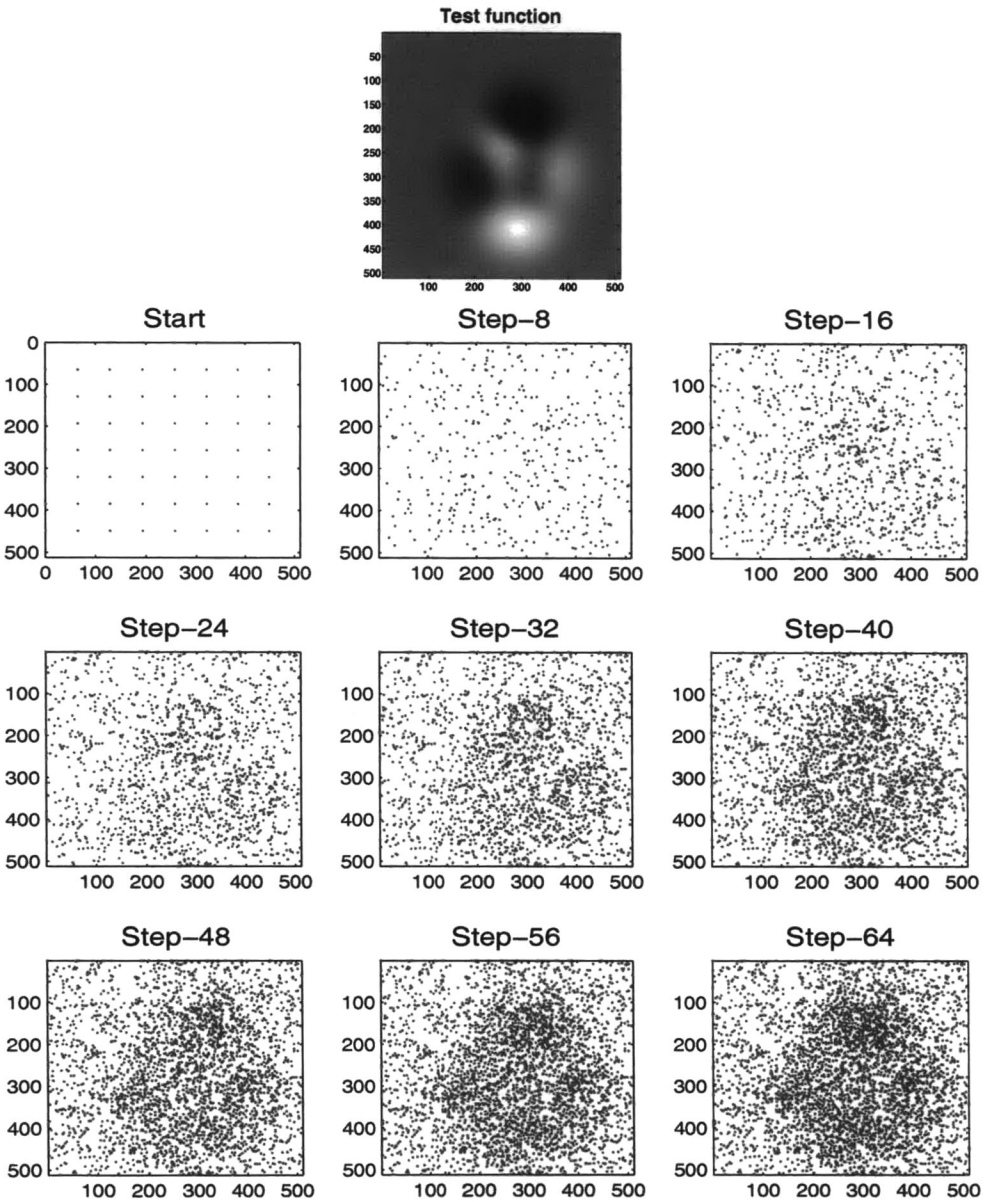


Figure 4.6: Intermediate steps from sampling a smooth test function with the active walker model using the following factors: $N_{\mathrm{aw}} = 64$, $N_{\mathrm{ub}} = 16$, $LSP = 0.4$, and $SSP = 0.06$. Total of 64 steps.

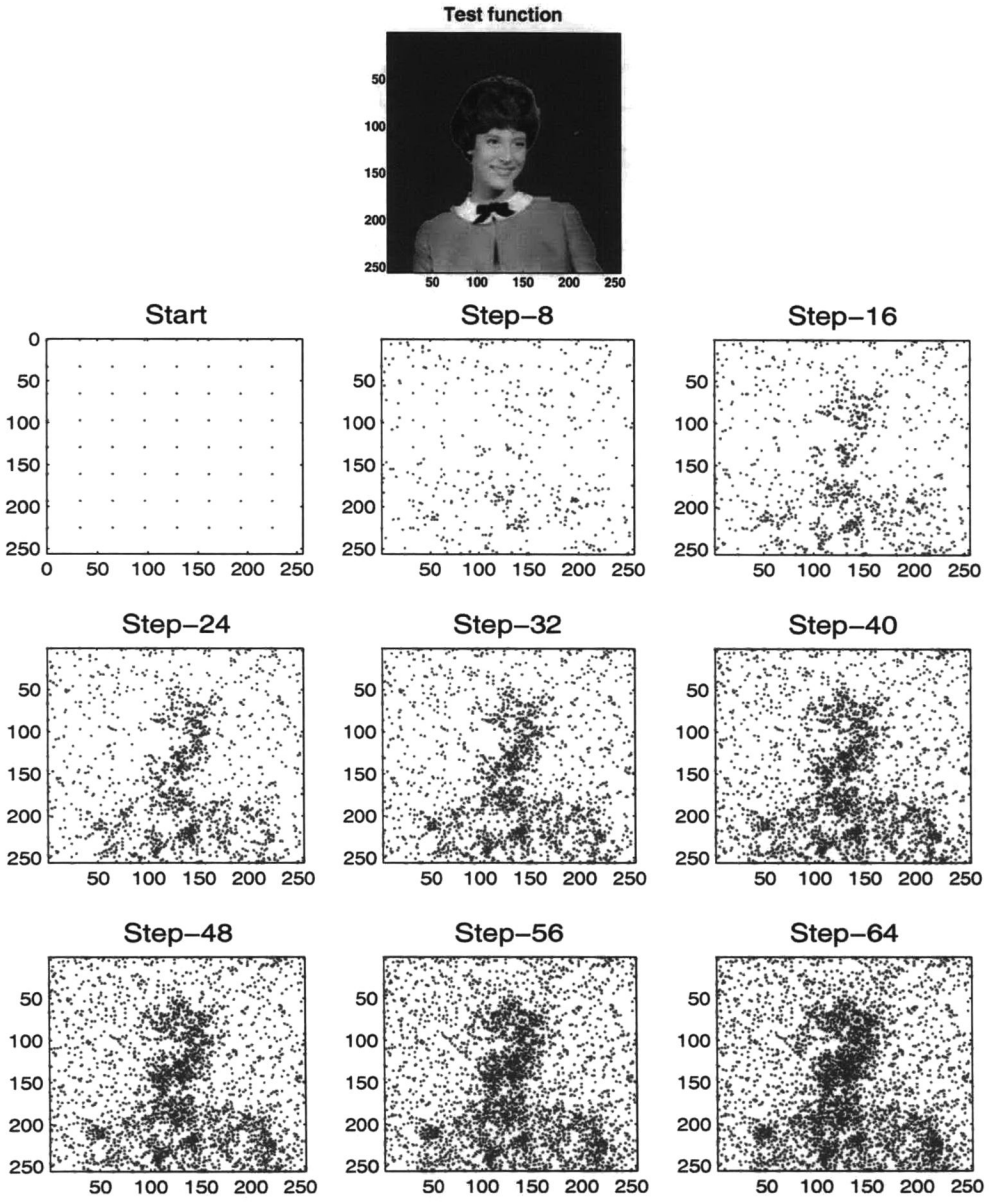


Figure 4.7: Intermediate steps from sampling a rapidly varying test function with the active walker model using the following factors: $N_{\mathrm{aw}} = 64$, $N_{\mathrm{ub}} = 16$, $LSP = 0.4$, and $SSP = 0.02$. Total of 64 steps.

Table 4.1: Performance of active walker model in sampling the smooth function, sorted by the frequency based measure.

No. of Active Walkers	No. of Bins	Long Step	Short step	CC with Freq. based Measure	CC with Ent. based Measure
100.00	16.00	0.40	0.06	0.60	0.93
144.00	16.00	0.40	0.06	0.60	0.93
144.00	16.00	0.35	0.06	0.59	0.92
144.00	16.00	0.40	0.08	0.59	0.91
4.00	16.00	0.35	0.06	0.58	0.91
4.00	16.00	0.40	0.08	0.58	0.91
144.00	16.00	0.35	0.04	0.58	0.92
64.00	16.00	0.40	0.06	0.57	0.93
100.00	16.00	0.35	0.08	0.57	0.87
64.00	16.00	0.40	0.08	0.57	0.91
100.00	16.00	0.35	0.06	0.57	0.92
100.00	16.00	0.40	0.08	0.57	0.90
64.00	16.00	0.35	0.06	0.57	0.92
144.00	16.00	0.30	0.04	0.57	0.91
4.00	16.00	0.40	0.06	0.56	0.92
64.00	16.00	0.30	0.06	0.56	0.90
4.00	16.00	0.30	0.06	0.56	0.90
144.00	16.00	0.35	0.08	0.56	0.88
144.00	32.00	0.35	0.08	0.56	0.89
144.00	16.00	0.40	0.04	0.56	0.92
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
1024.00	2.00	0.20	0.02	-0.26	0.14
64.00	16.00	0.25	0.10	-0.26	-0.35
1024.00	32.00	0.20	0.10	-0.30	-0.51
1024.00	16.00	0.20	0.10	-0.30	-0.48
144.00	256.00	0.20	0.10	-0.31	-0.70
64.00	256.00	0.20	0.10	-0.34	-0.74
100.00	256.00	0.20	0.10	-0.34	-0.70
4.00	256.00	0.20	0.10	-0.34	-0.73
144.00	8.00	0.20	0.10	-0.40	-0.74
64.00	8.00	0.20	0.10	-0.42	-0.76
4.00	8.00	0.20	0.10	-0.43	-0.76
100.00	8.00	0.20	0.10	-0.43	-0.73
100.00	32.00	0.20	0.10	-0.44	-0.80
4.00	32.00	0.20	0.10	-0.45	-0.83
64.00	16.00	0.20	0.10	-0.46	-0.79
64.00	32.00	0.20	0.10	-0.46	-0.81
144.00	32.00	0.20	0.10	-0.46	-0.81
4.00	16.00	0.20	0.10	-0.46	-0.80
144.00	16.00	0.20	0.10	-0.47	-0.77
100.00	16.00	0.20	0.10	-0.47	-0.80

Table 4.2: Performance of active walker model in sampling the smooth function, sorted by the entropy based measure.

No. of Active Walkers	No. of Bins	Long step	Short step	CC with Freq. based Measure	CC with Ent. based Measure
144.00	16.00	0.40	0.06	0.60	0.93
64.00	16.00	0.40	0.06	0.57	0.93
100.00	16.00	0.40	0.06	0.60	0.93
100.00	16.00	0.35	0.06	0.57	0.92
144.00	16.00	0.40	0.04	0.56	0.92
64.00	16.00	0.35	0.06	0.57	0.92
144.00	16.00	0.35	0.06	0.59	0.92
4.00	16.00	0.40	0.06	0.56	0.92
144.00	32.00	0.40	0.06	0.56	0.92
100.00	16.00	0.40	0.04	0.53	0.92
144.00	16.00	0.35	0.04	0.58	0.92
4.00	16.00	0.35	0.06	0.58	0.91
144.00	16.00	0.30	0.04	0.57	0.91
100.00	16.00	0.35	0.04	0.53	0.91
4.00	32.00	0.40	0.08	0.56	0.91
144.00	32.00	0.40	0.04	0.52	0.91
64.00	32.00	0.40	0.08	0.56	0.91
64.00	16.00	0.40	0.04	0.52	0.91
64.00	32.00	0.40	0.06	0.53	0.91
64.00	16.00	0.40	0.08	0.57	0.91
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
4.00	2.00	0.20	0.10	-0.10	-0.46
1024.00	16.00	0.20	0.10	-0.30	-0.48
64.00	2.00	0.20	0.10	-0.13	-0.48
1024.00	32.00	0.20	0.10	-0.30	-0.51
144.00	256.00	0.20	0.10	-0.31	-0.70
100.00	256.00	0.20	0.10	-0.34	-0.70
4.00	256.00	0.20	0.10	-0.34	-0.73
100.00	8.00	0.20	0.10	-0.43	-0.73
64.00	256.00	0.20	0.10	-0.34	-0.74
144.00	8.00	0.20	0.10	-0.40	-0.74
4.00	8.00	0.20	0.10	-0.43	-0.76
64.00	8.00	0.20	0.10	-0.42	-0.76
144.00	16.00	0.20	0.10	-0.47	-0.77
64.00	16.00	0.20	0.10	-0.46	-0.79
100.00	16.00	0.20	0.10	-0.47	-0.80
100.00	32.00	0.20	0.10	-0.44	-0.80
4.00	16.00	0.20	0.10	-0.46	-0.80
144.00	32.00	0.20	0.10	-0.46	-0.81
64.00	32.00	0.20	0.10	-0.46	-0.81
4.00	32.00	0.20	0.10	-0.45	-0.83

Table 4.3: Performance of active walker model in sampling the rapidly varying function, sorted by the frequency based measure.

No. of Active Walkers	No. of Bins	Long Step	Short step	CC with Freq. based Measure	CC with Ent. based Measure
4.00	16.00	0.40	0.02	0.72	0.88
4.00	16.00	0.35	0.02	0.69	0.86
64.00	16.00	0.40	0.02	0.68	0.87
4.00	32.00	0.40	0.02	0.68	0.86
4.00	16.00	0.30	0.02	0.67	0.85
64.00	32.00	0.40	0.02	0.66	0.89
64.00	16.00	0.35	0.02	0.66	0.86
4.00	32.00	0.35	0.02	0.65	0.84
64.00	16.00	0.30	0.02	0.64	0.85
100.00	16.00	0.40	0.02	0.63	0.86
64.00	32.00	0.35	0.02	0.62	0.86
100.00	16.00	0.35	0.02	0.62	0.85
4.00	16.00	0.25	0.02	0.61	0.82
4.00	8.00	0.40	0.02	0.61	0.80
144.00	16.00	0.40	0.02	0.61	0.84
144.00	16.00	0.35	0.02	0.59	0.84
4.00	8.00	0.35	0.02	0.59	0.79
100.00	32.00	0.40	0.02	0.59	0.85
4.00	8.00	0.30	0.02	0.59	0.79
64.00	16.00	0.25	0.02	0.59	0.82
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
1024.00	256.00	0.20	0.10	-0.18	-0.29
1024.00	32.00	0.20	0.10	-0.22	-0.38
1024.00	8.00	0.20	0.10	-0.23	-0.41
1024.00	16.00	0.20	0.10	-0.24	-0.43
144.00	256.00	0.20	0.10	-0.31	-0.56
4.00	256.00	0.20	0.10	-0.32	-0.55
64.00	256.00	0.20	0.10	-0.34	-0.56
4.00	8.00	0.20	0.10	-0.34	-0.62
100.00	256.00	0.20	0.10	-0.35	-0.58
64.00	8.00	0.20	0.10	-0.36	-0.64
100.00	8.00	0.20	0.10	-0.37	-0.63
144.00	8.00	0.20	0.10	-0.37	-0.65
144.00	32.00	0.20	0.10	-0.41	-0.69
144.00	16.00	0.20	0.10	-0.41	-0.69
4.00	16.00	0.20	0.10	-0.42	-0.68
100.00	16.00	0.20	0.10	-0.42	-0.67
4.00	32.00	0.20	0.10	-0.43	-0.68
100.00	32.00	0.20	0.10	-0.43	-0.68
64.00	16.00	0.20	0.10	-0.44	-0.69
64.00	32.00	0.20	0.10	-0.45	-0.70

Table 4.4: Performance of active walker model in sampling the rapidly varying function, sorted by the entropy based measure.

No. of Active Walkers	No. of Bins	Long Step	Short step	CC with Freq. based Measure	CC with Ent. based Measure
64.00	32.00	0.40	0.02	0.66	0.89
4.00	16.00	0.40	0.02	0.72	0.88
64.00	16.00	0.40	0.02	0.68	0.87
4.00	16.00	0.35	0.02	0.69	0.86
64.00	32.00	0.35	0.02	0.62	0.86
4.00	32.00	0.40	0.02	0.68	0.86
64.00	16.00	0.35	0.02	0.66	0.86
100.00	16.00	0.40	0.02	0.63	0.86
4.00	16.00	0.30	0.02	0.67	0.85
100.00	32.00	0.40	0.02	0.59	0.85
64.00	16.00	0.30	0.02	0.64	0.85
100.00	16.00	0.35	0.02	0.62	0.85
144.00	16.00	0.40	0.02	0.61	0.84
144.00	32.00	0.40	0.02	0.57	0.84
4.00	32.00	0.35	0.02	0.65	0.84
100.00	32.00	0.35	0.02	0.57	0.84
64.00	32.00	0.30	0.02	0.58	0.84
144.00	16.00	0.35	0.02	0.59	0.84
100.00	16.00	0.30	0.02	0.57	0.83
144.00	32.00	0.35	0.02	0.55	0.83
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
144.00	2.00	0.20	0.10	-0.11	-0.39
100.00	2.00	0.20	0.10	-0.13	-0.40
1024.00	8.00	0.20	0.10	-0.23	-0.41
1024.00	16.00	0.20	0.10	-0.24	-0.43
4.00	256.00	0.20	0.10	-0.32	-0.55
144.00	256.00	0.20	0.10	-0.31	-0.56
64.00	256.00	0.20	0.10	-0.34	-0.56
100.00	256.00	0.20	0.10	-0.35	-0.58
4.00	8.00	0.20	0.10	-0.34	-0.62
100.00	8.00	0.20	0.10	-0.37	-0.63
64.00	8.00	0.20	0.10	-0.36	-0.64
144.00	8.00	0.20	0.10	-0.37	-0.65
100.00	16.00	0.20	0.10	-0.42	-0.67
4.00	32.00	0.20	0.10	-0.43	-0.68
100.00	32.00	0.20	0.10	-0.43	-0.68
4.00	16.00	0.20	0.10	-0.42	-0.68
144.00	32.00	0.20	0.10	-0.41	-0.69
144.00	16.00	0.20	0.10	-0.41	-0.69
64.00	16.00	0.20	0.10	-0.44	-0.69
64.00	32.00	0.20	0.10	-0.45	-0.70

Also for this application, using extreme values of number of bins, such as N_{ub}= 2, or 256 produces poor results. This is because dividing the samples in to a very small number like two does not provide adequate resolution for the sample values, and therefore makes the basis of the ASHE algorithm irrelevant. Using a very high value like 256 means that it takes a longer time to form a profile in the histogram on which that ASHE is based. For this application, we obtain good results for values of N_{ub}= 16, and 32. This is to be chosen based on the expected number of unique samples in the function being sampled.

The values of LSP and SSP are the more crucial factors for the performance of this model. We always obtain poor performance when the values of LSP and SSP are comparable. This is expected since the movement of the active walkers under this condition does not reflect the effect of the fitness criterion, and is essentially random. Values of $LSP \geq 0.3$ result in good performance since it is an appreciable movement of the active walker away from its present vicinity in response to change in the fitness criterion. The SSP value is the most crucial factor. Values of SSP= 0.06 give the best result for sampling the smooth function adaptively. The stated LSP value is appropriate for all scenarios. The chosen SSP value will depend on whether a function is smooth or varying rapidly.

All the discussions for the experiments with the smooth function, are also valid for the results for the rapidly varying function shown in Tables 4.3 and 4.4. The only difference is in the crucial factor, SSP. Note here that the best performance is obtained for $SSP = 0.02$, compared to $SSP = 0.06$ for the smooth function. This is accounted for by the fact that the function in the latter case varies rapidly, and a small step results in much larger changes in sample values. Note that this is the only difference in sampling the two types of functions. This is important because algorithms like the active walker model

usually suffer the drawback of being *ad-hoc*. That is, they have to be customized for every unique purpose. Our results show that "rules of thumb" can be established for determining the other factors in the active walker model. Customization of the model for any purpose only requires minimal prior knowledge on whether a function to be sampled is smooth or varies rapidly. This prior information is available in many cases.

Figures 4.8 to 4.11 show the stages of the adaptive sampling process for the two test functions, using the active walker model. We show examples of the good (Figures 4.8 and 4.10), and poor (Figures 4.9 and 4.11) performances as indicated on the tables. In the examples with the poor performance, the sample density show a random pattern, supporting the argument given for the effect of the LSP, and SSP factors. Only intermediate steps are shown. The examples showing good performance show sample densities indicative of the regions of complexities in the function.

4.4 Analysis of the Ant Model

Here, we also experiment with the same test functions, and obtained the same number of samples as described in Section 4.3. We identify four key factors that may affect the performance of the ant model described in Section 3.2.1.

These are:

1. Number of ants (N_{as}). This will determine the number of foraging trips taken in the adaptive sampling process. Starting with n ants means that $(4,096/n) - 1$ trips will be taken.

2. Number of bins N_{ub} in the histogram to be equalized. Same explanation under the active walker model holds here.

3. The range of the effect of the deposited pheromone (RPH).

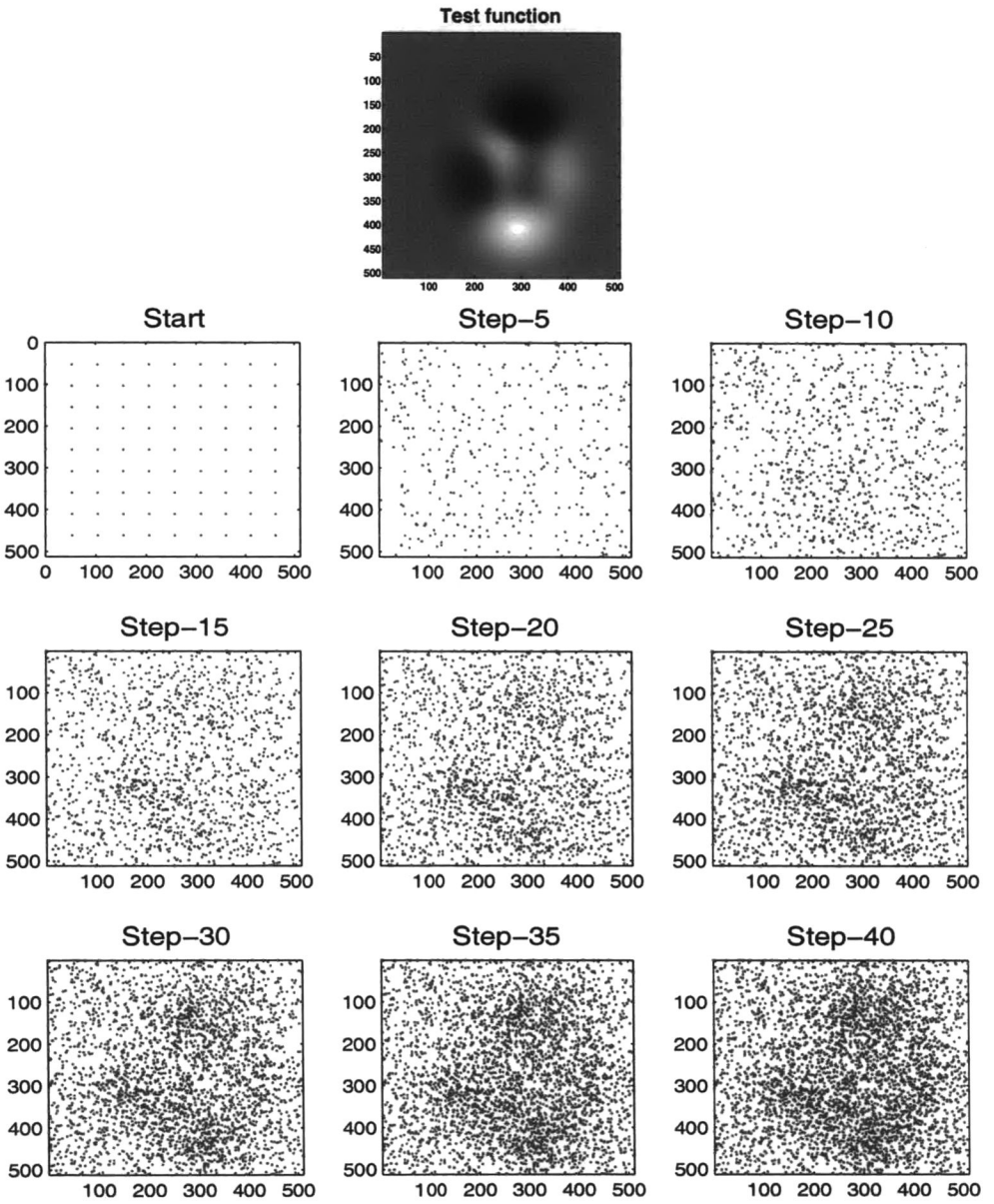


Figure 4.8: Intermediate steps from sampling a smooth test function with the active walker model. The good performance is recorded by using the following factors: $N_{\text{aw}} = 100$, $N_{\text{ub}} = 16$, $LSP = 0.4$, and $SSP = 0.06$. Performance measures = 0.60/0.93. Total of 40 steps.

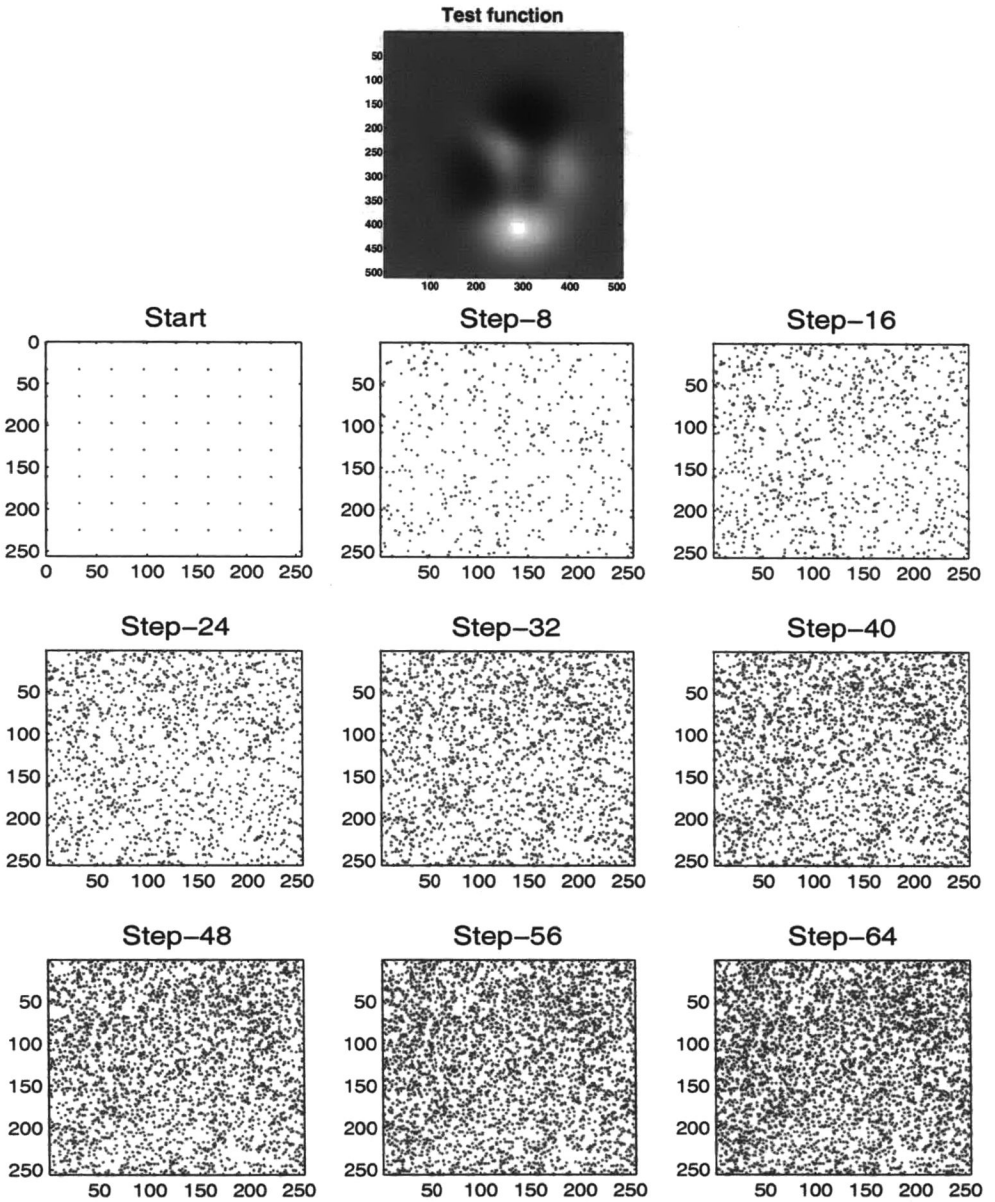


Figure 4.9: Intermediate steps from sampling a smooth test function with the active walker model. The poor performance is recorded by using the following factors: N_{aw} = 100, N_{ub} = 16, LSP = 0.2, and SSP = 0.1. Performance measures = -0.47/-0.80. Total of 40 steps.

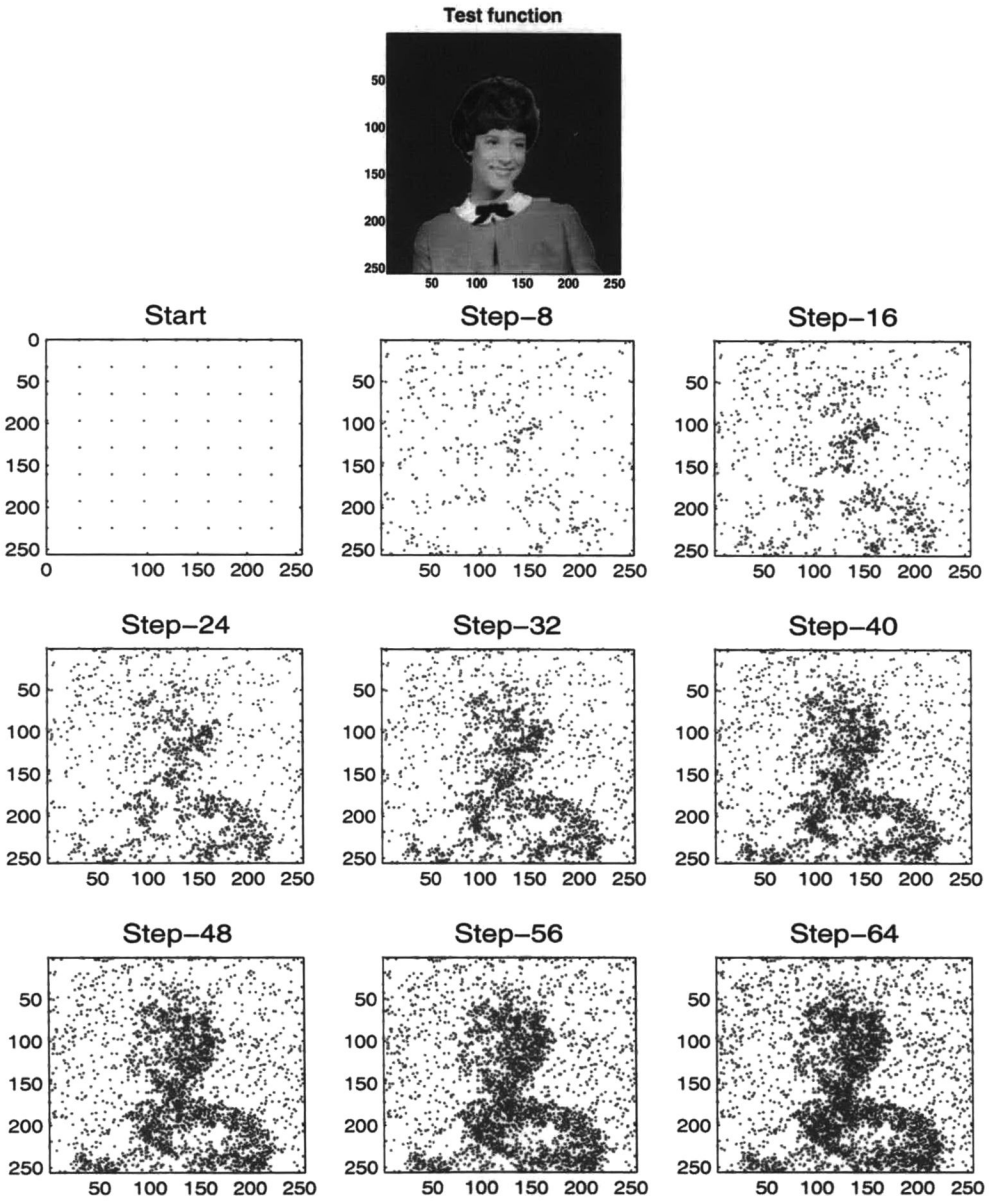


Figure 4.10: Intermediate steps from sampling a rapidly varying function with the active walker model. The good performance is recorded by using the following factors: $N_{\mathrm{aw}} = 64$, $N_{\mathrm{ub}} = 32$, $LSP = 0.4$, and $SSP = 0.02$. Performance measure = 0.66/0.89. Total of 64 steps.

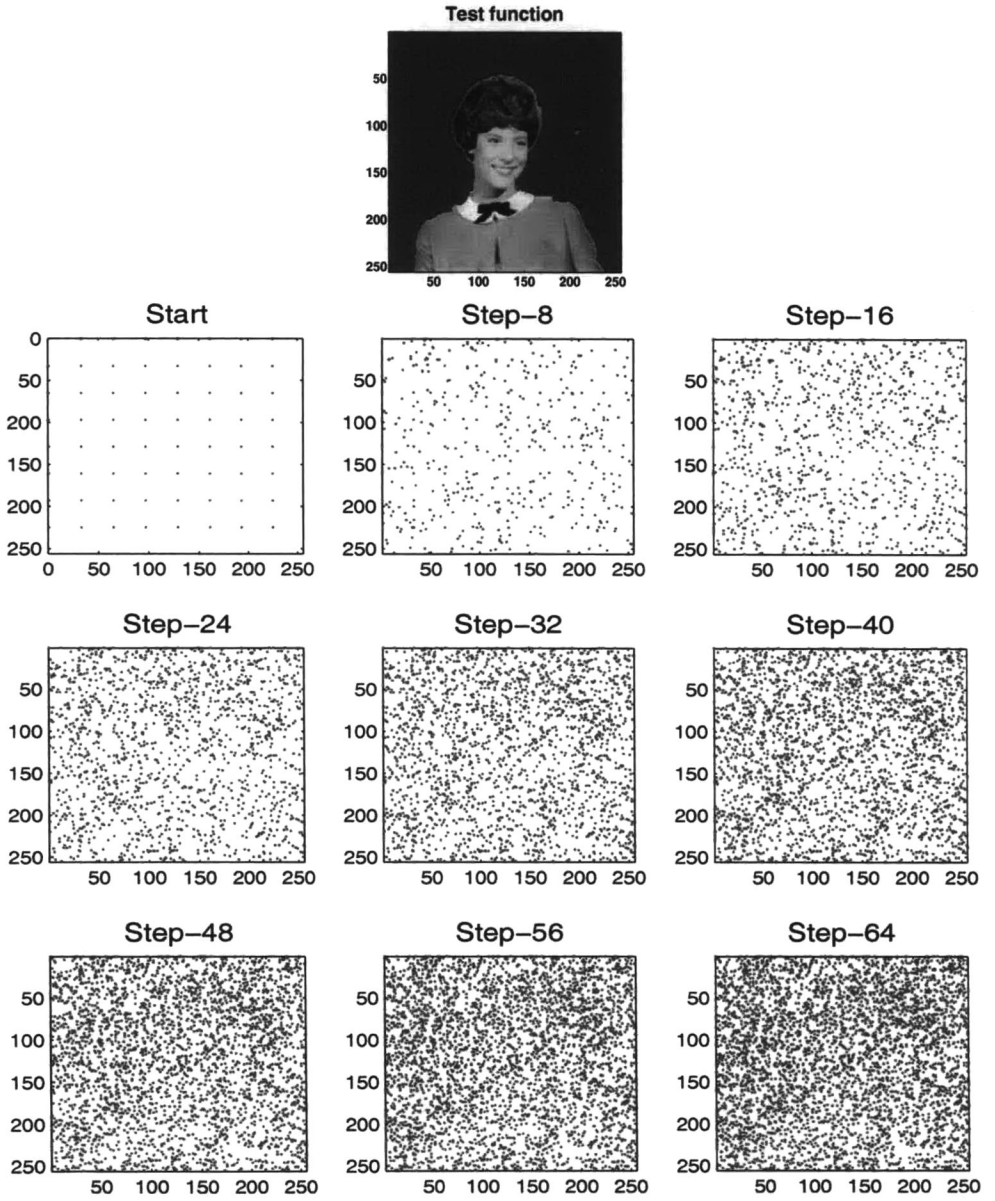


Figure 4.11: Intermediate steps from sampling a rapidly varying function with the active walker model. The poor performance is recorded by using the following factors: N_{aw} = 64, N_{ub} = 32, LSP = 0.2, and SSP = 0.2. Performance measure = 0.66/0.89. Total of 64 steps.

4. The range an ant can move in its neighborhood for foraging (RFO).

We experiment with the following values of these factors:

$N_{\text{as}} = \{4,\ 64,\ 100,\ 144,\ 1,024\}$

$N_{\text{ub}} = \{2,\ 8,\ 16,\ 32,\ 256\}$

$RPH = \{0.02,\ 0.05,\ 0.1,\ 0.12,\ 0.15,\ 0.2,\ 0.25\}$

$RFO = \{0.02,\ 0.05,\ 0.08,\ 0.1,\ 0.12,\ 0.15,\ 0.2\}$

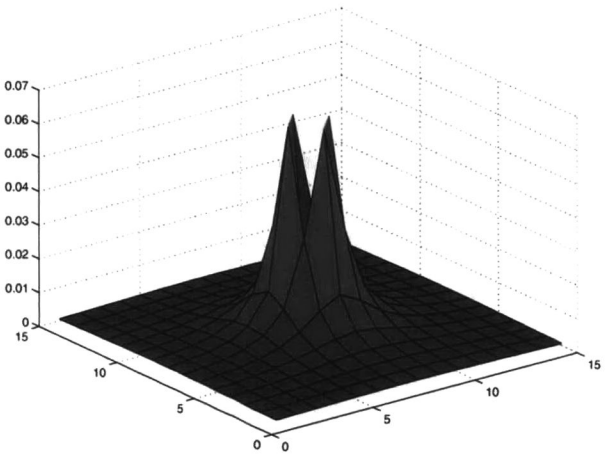


Figure 4.12: Model to simulate reduction of pheromone concentration as a function of the distance away from source. Concentration $\propto 1/d^2$.

The last two factors, RPH, and RFO are functions of the size of the sampled space. That is, they are determined by multiplying these numbers with the vector containing the spaces' dimensions. The RPH is made to fade away from the point of deposit at a rate of square of the distance d. That is the concentration $\propto 1/d^2$. The concentration, represented as probabilities, are normalized, and added to the existing value in a non-zero probability location. Figure 4.12 shows an example of how the concentration fades away with distance from the source. The central pixel is set to a probability of zero since that location has already been sampled.

Figures 4.13 and 4.14 show the change in the pheromone concentration, modeling the

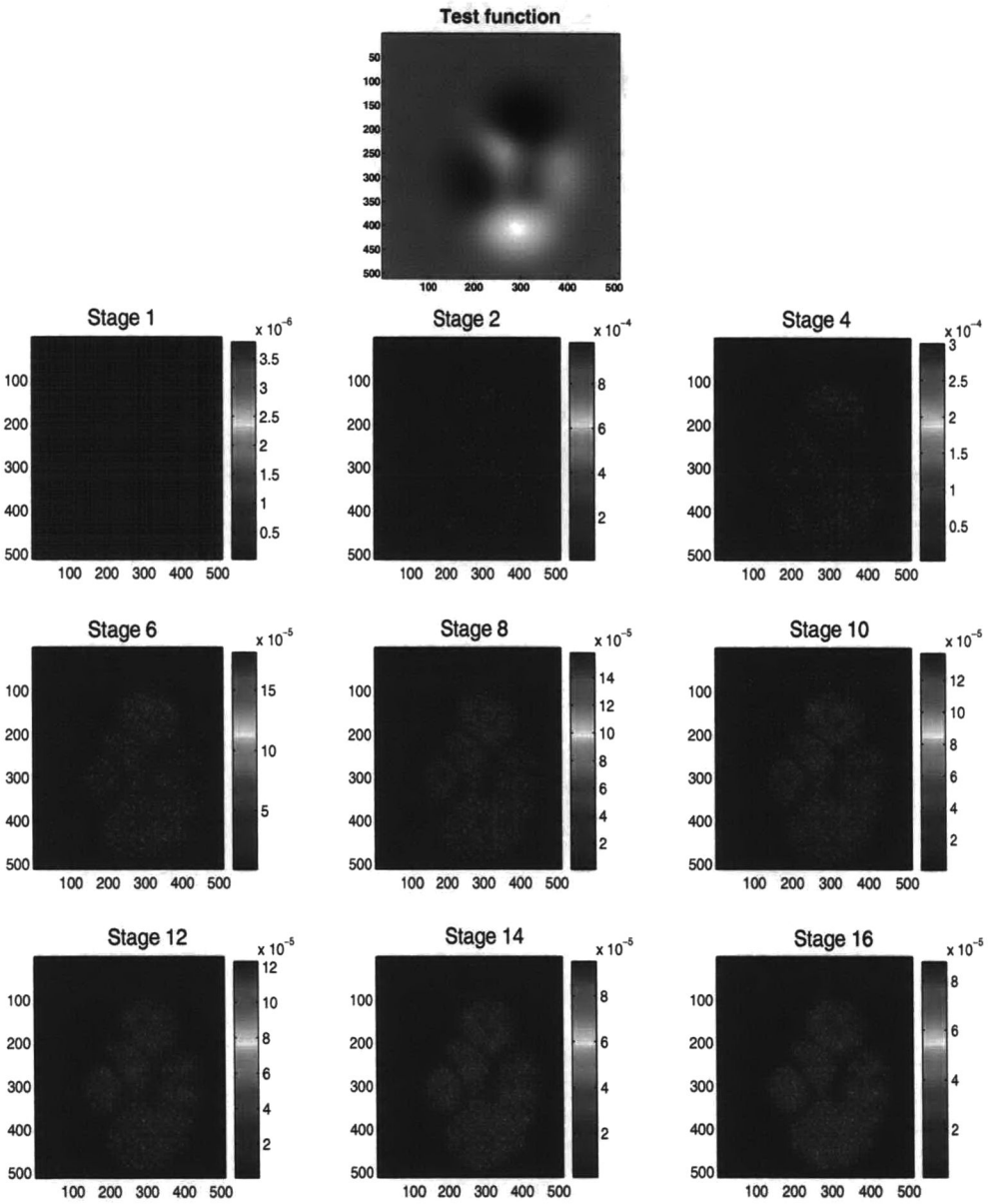


Figure 4.13: Pheromone concentration/probability change from intermediate steps in the sampling of a smooth function. Ant model used with the following factors: $N_{\text{as}} = 256$, $N_{\text{ub}} = 8$, $RPH = 0.12$, and $RFO = 0.12$. Total of 16 foraging trips by each ant.

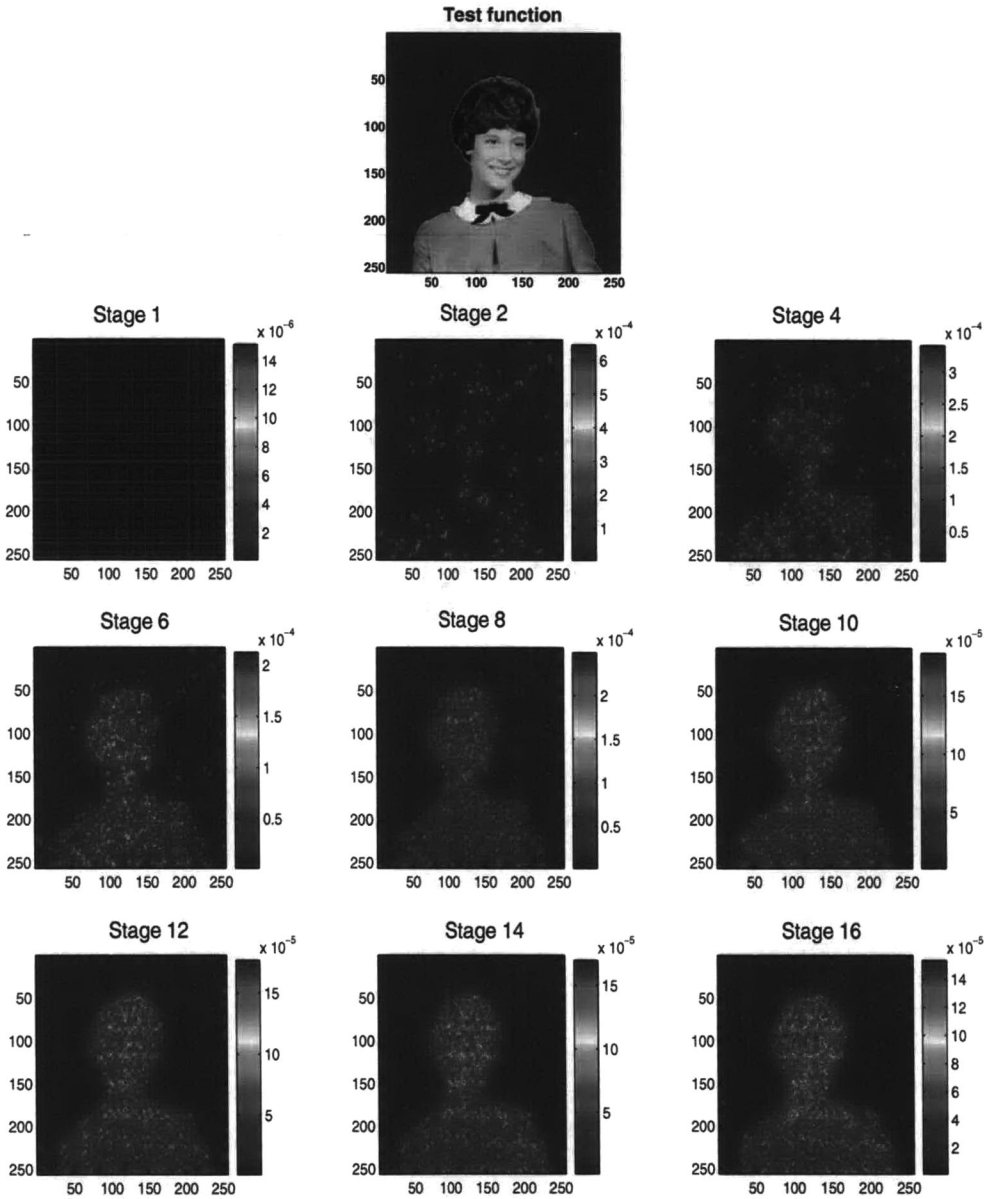


Figure 4.14: Pheromone concentration/probability change from intermediate steps in the sampling of a rapidly varying function. Ant model used with the following factors: $N_{\text{as}} = 256$, $N_{\text{ub}} = 32$, $RPH = 0.25$, and $RFO = 0.12$. Total of 16 foraging trips by each ant.

probabilities in the sampled space. Note how the probabilities change to reflect the levels of complexity in the sampled functions. Only intermediate steps are shown because of the space constraint.

Tables 4.5 to 4.8 show similarly sorted results as in Section 4.3. The higher correlation between the sample density and the entropy measure was also noted, and explained in the active walker model analysis discussion. The good correlation between both objective measures indicates that they essentially convey the same information. Tables 4.5 and 4.6 show the sorted results for the experiments with the smooth function. The results indicate that too few foraging trips yield a poor sample distribution. Generally, the poor performance results from using too many ants $N_{\text{as}} = 1,024$, each taking only three foraging trips. The reason for this is similar to that stated for the active walker model. There are too few steps to enable the feedback from the ants have an effect. The sampling under this condition is thus near random. Good performances are recorded for 15 or more foraging trips, resulting from employing $N_{\text{as}} \leq 256$. The N_{ub} also has similar effects as in the case of the active walker model. Good performance is recorded for values of at least eight bins in the histogram. Extreme N_{ub} values of 2 or 256 also result in poor performance for reasons stated earlier. No clear trend can be deciphered from the tables about the values representing the range of the effect of the deposited pheromone RPH. A clear correlation is needed between all the factors and the performance criteria, in order to establish appropriate values for the factors. Values of rfo ≥ 0.1 result in good performance.

Similar results are recorded from the experiments with the rapidly varying function shown in Tables 4.7 and 4.8. All the discussions for the experiments with the smooth function, are also valid here.

Figures 4.15 to 4.18 show the stages of the adaptive sampling process for the two test

Table 4.5: Performance of the ant model in sampling the smooth function, sorted by the frequency based measure.

No. of ANTs	No. of Bins	Range of Pheromone	Range of Foraging(1 step)	CC with Freq. based Measure	CC with Ent. based Measure
256.00	8.00	0.15	0.08	0.64	0.90
256.00	8.00	0.02	0.08	0.64	0.89
256.00	8.00	0.05	0.08	0.64	0.89
256.00	8.00	0.10	0.08	0.64	0.90
256.00	8.00	0.20	0.08	0.64	0.91
256.00	8.00	0.12	0.08	0.64	0.90
256.00	8.00	0.02	0.10	0.63	0.92
256.00	8.00	0.25	0.08	0.63	0.92
256.00	8.00	0.15	0.10	0.63	0.93
256.00	8.00	0.10	0.10	0.63	0.93
256.00	8.00	0.05	0.10	0.63	0.92
256.00	8.00	0.12	0.10	0.62	0.93
144.00	8.00	0.05	0.08	0.62	0.92
64.00	8.00	0.10	0.05	0.62	0.89
144.00	8.00	0.12	0.05	0.62	0.84
144.00	8.00	0.10	0.05	0.62	0.84
144.00	8.00	0.05	0.05	0.62	0.81
256.00	8.00	0.05	0.12	0.62	0.93
144.00	8.00	0.02	0.08	0.62	0.92
256.00	8.00	0.25	0.10	0.62	0.93
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
256.00	2.00	0.10	0.02	-0.08	0.10
1024.00	2.00	0.20	0.08	-0.09	0.36
256.00	2.00	0.02	0.02	-0.10	0.01
144.00	2.00	0.02	0.02	-0.10	0.05
1024.00	2.00	0.02	0.05	-0.11	0.06
256.00	2.00	0.05	0.02	-0.11	0.05
1024.00	2.00	0.05	0.10	-0.11	0.40
1024.00	2.00	0.10	0.10	-0.12	0.40
1024.00	2.00	0.12	0.05	-0.12	0.09
256.00	2.00	0.12	0.02	-0.12	0.09
1024.00	2.00	0.15	0.05	-0.12	0.10
1024.00	2.00	0.20	0.05	-0.16	0.13
1024.00	2.00	0.15	0.08	-0.16	0.28
1024.00	2.00	0.02	0.10	-0.17	0.36
1024.00	2.00	0.05	0.05	-0.18	0.02
1024.00	2.00	0.10	0.05	-0.19	0.09
1024.00	2.00	0.10	0.08	-0.21	0.23
1024.00	2.00	0.12	0.08	-0.21	0.24
1024.00	2.00	0.02	0.08	-0.24	0.17
1024.00	2.00	0.05	0.08	-0.24	0.19

Table 4.6: Performance of the ant model in sampling the smooth function, sorted by the entropy based measure.

No. of ANTs	No. of Bins	Range of Pheromone	Range of Foraging(1 step)	CC with Freq. based Measure	CC with Ent. based Measure
256.00	8.00	0.12	0.12	0.61	0.94
256.00	8.00	0.02	0.12	0.62	0.93
256.00	8.00	0.25	0.12	0.60	0.93
144.00	8.00	0.25	0.10	0.57	0.93
256.00	8.00	0.20	0.10	0.61	0.93
256.00	8.00	0.10	0.12	0.62	0.93
256.00	8.00	0.20	0.12	0.61	0.93
256.00	8.00	0.15	0.12	0.61	0.93
144.00	8.00	0.12	0.10	0.57	0.93
144.00	8.00	0.10	0.10	0.59	0.93
144.00	8.00	0.20	0.10	0.57	0.93
144.00	16.00	0.02	0.12	0.53	0.93
144.00	8.00	0.12	0.12	0.54	0.93
144.00	8.00	0.02	0.12	0.55	0.93
144.00	8.00	0.15	0.10	0.57	0.93
256.00	8.00	0.25	0.10	0.62	0.93
256.00	8.00	0.10	0.15	0.56	0.93
256.00	8.00	0.05	0.12	0.62	0.93
144.00	8.00	0.15	0.12	0.54	0.93
144.00	8.00	0.20	0.12	0.54	0.93
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
1024.00	8.00	0.05	0.02	0.08	0.04
1024.00	16.00	0.15	0.02	-0.00	0.04
1024.00	256.00	0.02	0.02	-0.01	0.04
1024.00	8.00	0.02	0.02	0.04	0.03
1024.00	2.00	0.10	0.02	-0.06	0.03
1024.00	32.00	0.05	0.02	-0.01	0.03
1024.00	32.00	0.10	0.02	-0.03	0.03
1024.00	2.00	0.20	0.02	-0.02	0.02
1024.00	2.00	0.05	0.05	-0.18	0.02
1024.00	256.00	0.25	0.02	-0.03	0.02
1024.00	32.00	0.20	0.02	0.00	0.02
1024.00	256.00	0.20	0.02	-0.01	0.02
256.00	2.00	0.02	0.02	-0.10	0.01
1024.00	2.00	0.25	0.02	-0.05	0.00
1024.00	16.00	0.05	0.02	-0.00	-0.00
1024.00	2.00	0.15	0.02	-0.08	-0.00
1024.00	2.00	0.02	0.02	-0.05	-0.01
1024.00	2.00	0.12	0.02	-0.06	-0.01
1024.00	32.00	0.02	0.02	-0.03	-0.02
1024.00	2.00	0.05	0.02	-0.04	-0.03

Table 4.7: Performance of the ant model in sampling the rapidly varying function, sorted by the frequency based measure.

No. of ANTs	No. of Bins	Range of Pheromone	Range of Foraging(1 step)	CC with Freq. based Measure	CC with Ent. based Measure
256.00	32.00	0.25	0.12	0.78	0.89
256.00	256.00	0.12	0.12	0.78	0.88
256.00	32.00	0.20	0.12	0.78	0.89
256.00	32.00	0.15	0.12	0.78	0.89
256.00	32.00	0.10	0.12	0.78	0.89
256.00	256.00	0.25	0.12	0.78	0.88
256.00	16.00	0.15	0.12	0.77	0.90
256.00	256.00	0.10	0.12	0.77	0.88
256.00	256.00	0.20	0.12	0.77	0.88
256.00	256.00	0.05	0.12	0.77	0.89
256.00	32.00	0.05	0.12	0.77	0.89
256.00	32.00	0.12	0.12	0.77	0.89
256.00	16.00	0.25	0.12	0.77	0.90
256.00	32.00	0.02	0.12	0.77	0.89
256.00	16.00	0.20	0.12	0.77	0.90
256.00	16.00	0.20	0.10	0.77	0.88
256.00	16.00	0.25	0.10	0.77	0.88
256.00	256.00	0.02	0.12	0.77	0.88
256.00	256.00	0.10	0.15	0.77	0.87
256.00	32.00	0.12	0.10	0.77	0.88
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
1024.00	256.00	0.05	0.02	0.05	0.03
144.00	2.00	0.02	0.02	0.05	0.12
1024.00	256.00	0.02	0.02	0.04	0.06
1024.00	2.00	0.10	0.05	0.03	0.16
1024.00	32.00	0.02	0.02	0.02	0.02
1024.00	16.00	0.02	0.02	0.02	0.04
256.00	2.00	0.05	0.02	0.02	0.11
256.00	2.00	0.10	0.02	0.02	0.15
1024.00	2.00	0.20	0.02	0.02	0.05
1024.00	2.00	0.15	0.02	0.02	0.05
1024.00	8.00	0.05	0.02	0.02	0.02
1024.00	32.00	0.05	0.02	0.01	0.02
1024.00	8.00	0.02	0.02	0.01	0.02
1024.00	2.00	0.02	0.05	0.01	0.10
1024.00	2.00	0.02	0.02	-0.00	0.00
1024.00	2.00	0.12	0.02	-0.00	0.02
256.00	2.00	0.02	0.02	-0.01	0.03
1024.00	2.00	0.05	0.05	-0.02	0.07
1024.00	2.00	0.10	0.02	-0.02	-0.00
1024.00	2.00	0.05	0.02	-0.03	-0.01

Table 4.8: Performance of the ant model in sampling the rapidly varying function, sorted by the entropy based measure.

No. of ANTs	No. of Bins	Range of Pheromone	Range of Foraging(1 step)	CC with Freq. based Measure	CC with Ent. based Measure
144.00	16.00	0.05	0.12	0.73	0.91
144.00	16.00	0.02	0.12	0.72	0.91
144.00	16.00	0.12	0.12	0.74	0.91
144.00	8.00	0.05	0.12	0.71	0.91
144.00	8.00	0.10	0.12	0.72	0.91
144.00	16.00	0.15	0.12	0.73	0.90
144.00	32.00	0.12	0.12	0.74	0.90
144.00	8.00	0.15	0.10	0.73	0.90
144.00	16.00	0.10	0.12	0.73	0.90
144.00	8.00	0.20	0.12	0.72	0.90
144.00	16.00	0.10	0.10	0.74	0.90
256.00	16.00	0.05	0.12	0.77	0.90
144.00	16.00	0.25	0.12	0.73	0.90
144.00	8.00	0.02	0.12	0.70	0.90
144.00	32.00	0.20	0.12	0.73	0.90
144.00	32.00	0.15	0.12	0.74	0.90
144.00	8.00	0.15	0.12	0.71	0.90
144.00	8.00	0.12	0.12	0.71	0.90
144.00	32.00	0.05	0.12	0.74	0.90
144.00	16.00	0.20	0.12	0.73	0.90
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
1024.00	256.00	0.15	0.02	0.08	0.07
1024.00	2.00	0.05	0.05	-0.02	0.07
1024.00	256.00	0.12	0.02	0.06	0.06
1024.00	8.00	0.10	0.02	0.06	0.06
1024.00	256.00	0.10	0.02	0.06	0.06
1024.00	256.00	0.02	0.02	0.04	0.06
1024.00	16.00	0.05	0.02	0.09	0.06
1024.00	2.00	0.15	0.02	0.02	0.05
1024.00	2.00	0.20	0.02	0.02	0.05
1024.00	16.00	0.02	0.02	0.02	0.04
1024.00	256.00	0.05	0.02	0.05	0.03
256.00	2.00	0.02	0.02	-0.01	0.03
1024.00	2.00	0.12	0.02	-0.00	0.02
1024.00	8.00	0.05	0.02	0.02	0.02
1024.00	32.00	0.02	0.02	0.02	0.02
1024.00	8.00	0.02	0.02	0.01	0.02
1024.00	32.00	0.05	0.02	0.01	0.02
1024.00	2.00	0.02	0.02	-0.00	0.00
1024.00	2.00	0.10	0.02	-0.02	-0.00
1024.00	2.00	0.05	0.02	-0.03	-0.01

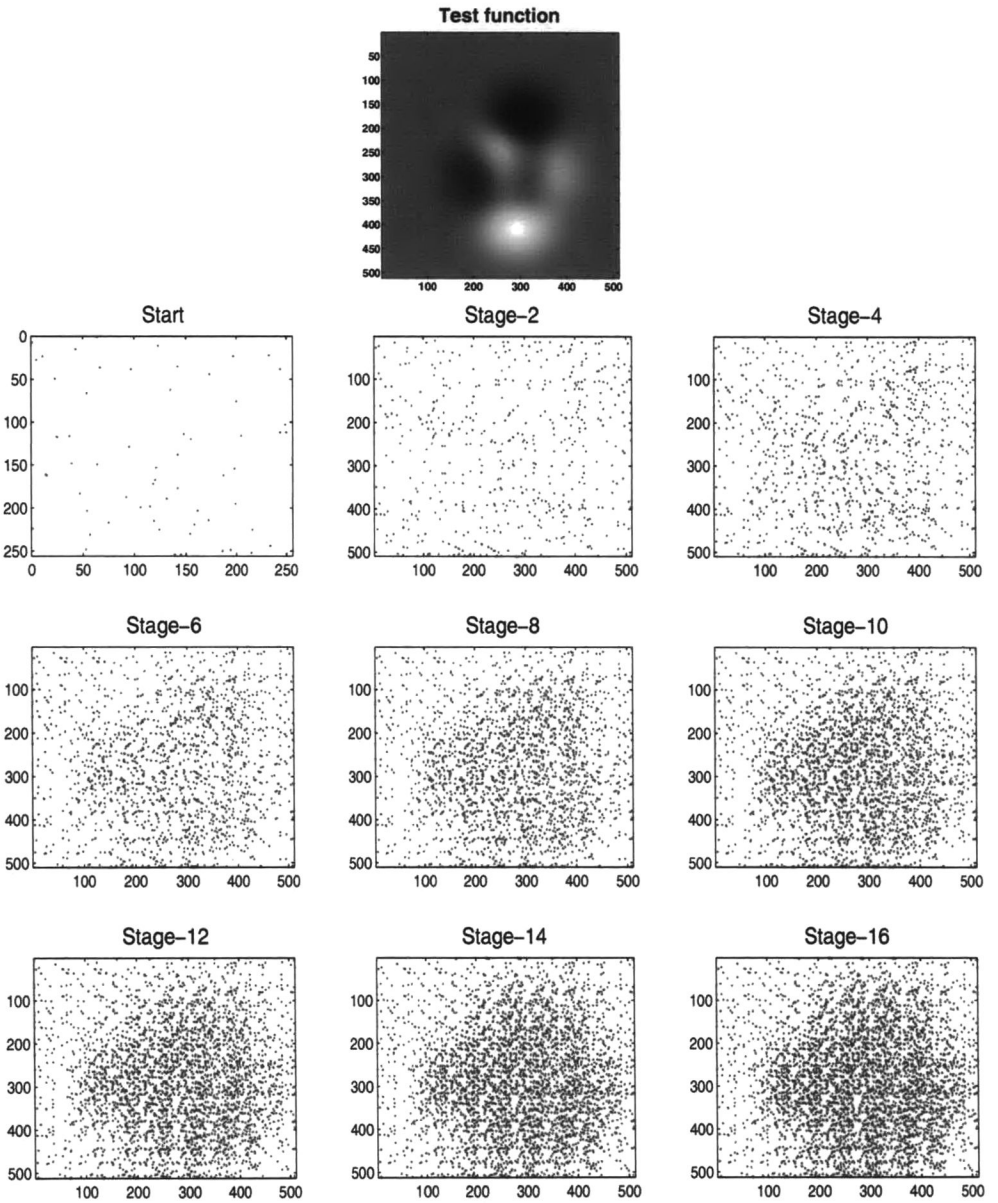


Figure 4.15: Intermediate stages from sampling a smooth test function with the ant model. The good performance is recorded by using the following factors: $N_{\text{as}} = 256$, $N_{\text{ub}} = 8$, $RPH = 0.12$, and $RFO = 0.12$. Performance measures = 0.60/0.91. Total of 16 foraging trips.

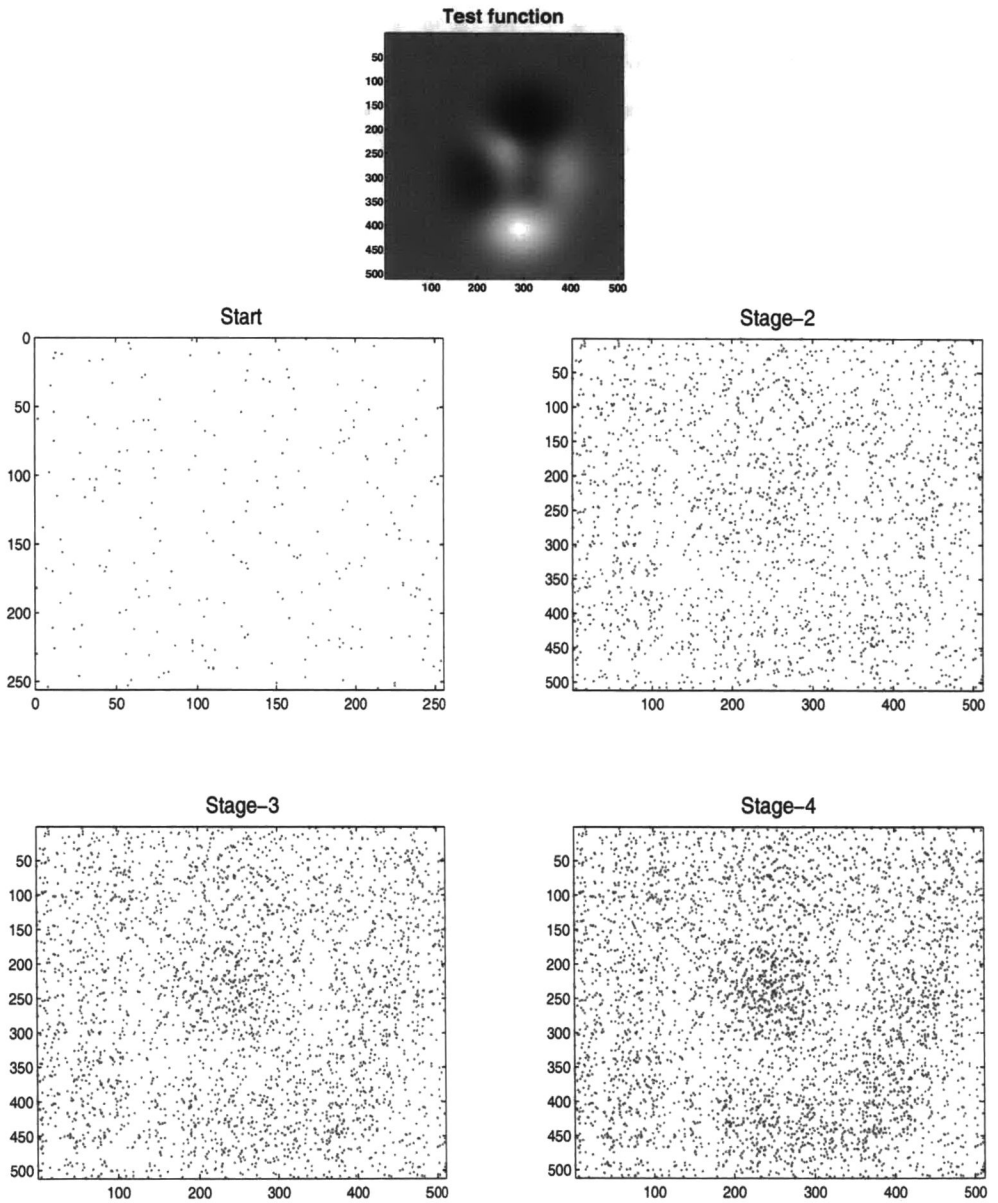


Figure 4.16: All the stages from sampling a smooth test function with the ant model. The poor performance is recorded by using the following factors: N_{as} = 1,024, N_{ub} = 2, RPH = 0.05, and RFO = 0.08. Performance measures = -0.24/0.19. Total of four foraging trips.

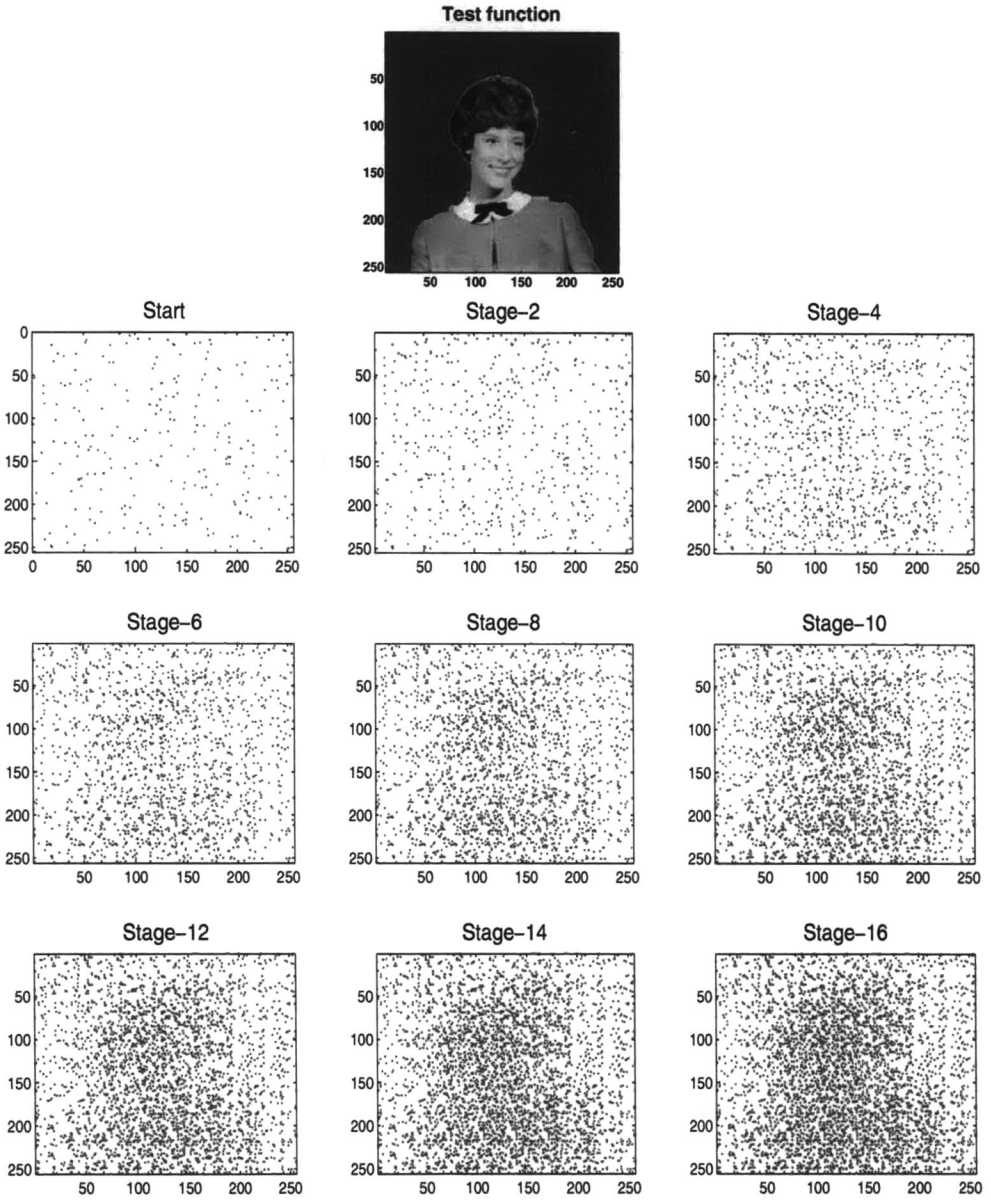


Figure 4.17: Intermediate stages from sampling a rapidly varying test function with the ant model. The good performance is recorded by using the following factors: $N_{\text{as}} = 256$, $N_{\text{ub}} = 32$, $RPH = 0.25$, and $RFO = 0.12$. Performance measures = 0.78/0.89. Total of 16 foraging trips.

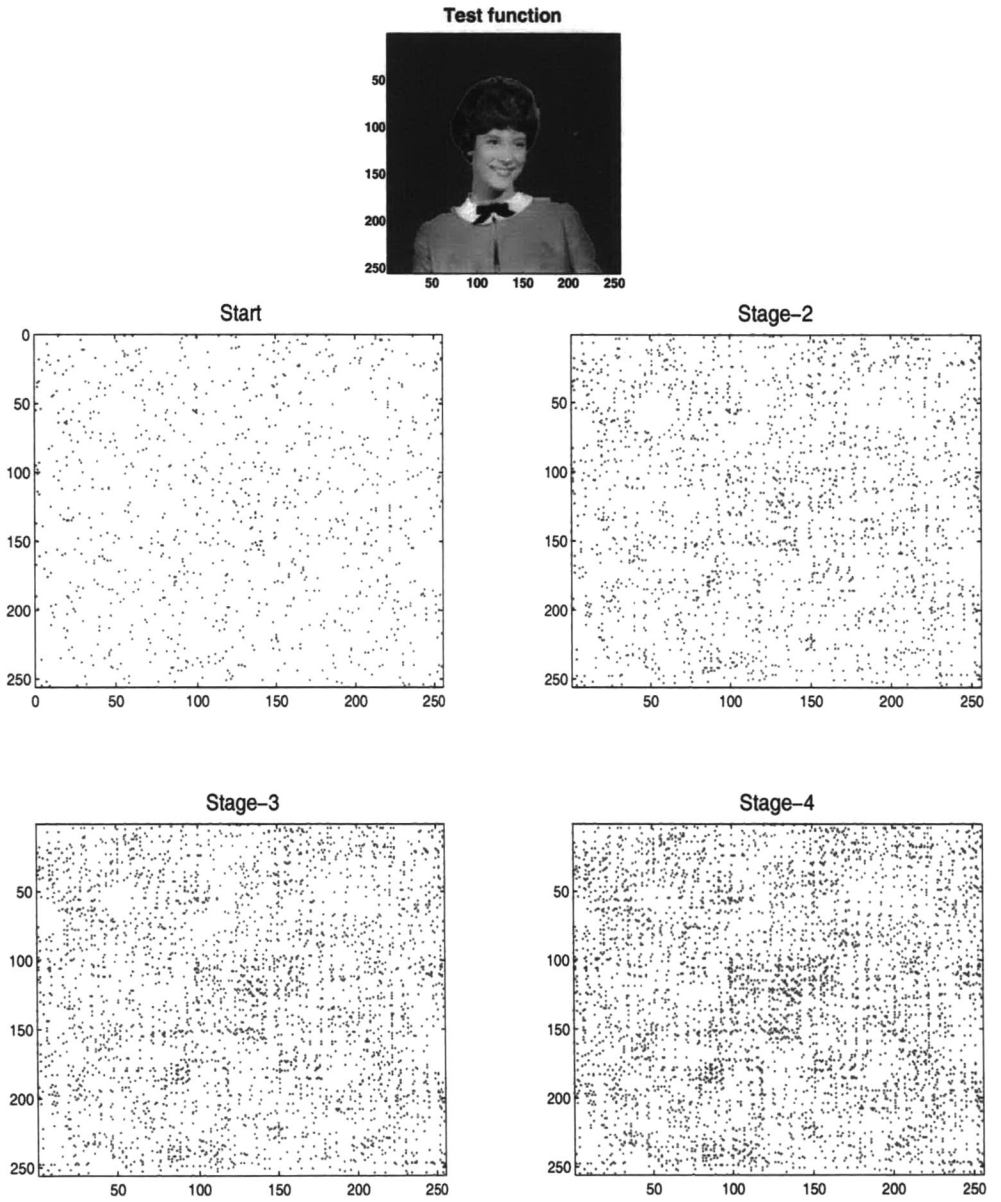


Figure 4.18: All the stages from sampling a rapidly varying test function with the ant model. The poor performance is recorded by using the following factors: N_{as} = 1,024, N_{ub} = 2, RPH = 0.05, and RFO = 0.02. Performance measures = -0.03/0.01. Total of four foraging trips.

functions, using the ant model. We show examples of the good, and poor performances as indicated on the tables. Only intermediate steps are shown in some of the cases.

4.5 Analysis of the Evolutionary Algorithm (EA) Model

The test functions, and the number of samples obtained using the EA model, are the same as in the two models considered earlier. We identify five key factors that may affect the performance of the EA model as described in Section 3.3.1. These are:

1. Size of starting population (SOP). This will determine the number of generations in the adaptive sampling process. Starting with a population of size n means that there will be $(4,096/n) - 1$ generations.

2. Number of offspring (N_{of}) from each parent.

3. Number of bins (N_{ub}) in the histogram to be equalized. Same explanation under the previously discussed models holds here.

4. Distance away from parent (DOP) or neighborhood where an offspring will reside.

5. The probability of survival of an offspring (POS).

We experiment with the following values of these factors:

$SOP = \{4,\ 64,\ 144,\ 1,024\}$

$N_{\text{of}} = \{1,\ 2,\ 4,\ 6\}$

$N_{\text{ub}} = \{4,\ 16,\ 32,\ 256\}$

$DOP = \{0.02,\ 0.05,\ 0.08,\ 0.1\}$

$POS = \{1,\ 0.9,\ 0.8,\ 0.7\}$

The DOP factor is a function of the size of the sampled space. That is, it is determined by multiplying the listed number with the vector containing the spaces' dimensions. A complete set of experiments included $1,024$ runs, resulting from all combinations of the factors. We ran 100 complete sets, requiring $102,400$ experiments.

The performance results from the experiments with the smooth function are sorted as in Section 4.3, and shown in Tables 4.9 to 4.12. For both test functions, the only factor that shows a general trend in indicating sampling performance is the N_{ub}. Extreme values of 4 and 256 generally result in the poor performance of the algorithm. Values of 16 and 32 result in the recorded good performance. The reason for this trend is the same as in the previously discussed models. It is important to note that this factor is important for all the employed models. Note that the EA algorithm is able to *converge* at a much faster rate than the other two models. This is indicated by the fact that a starting population of $1,024$ may still result in a good performance. As shown on the tables, this is only recorded when parents have multiple offspring, that is, $N_{\text{of}} \geq 2$. This rapid convergence is true for the smooth function in particular. The other four factors, do not show any unique correlation to the sampling performance. This is a significant drawback in applying the model in a practical application.

Figures 4.19 to 4.22 show the stages of the adaptive sampling process for the two test functions, using the EA model. We show examples of the good, and poor performances as indicated on the tables. As before, only intermediate steps are shown in some of the cases due to the space constraint.

Table 4.9: Performance of the Evolutionary Algorithm model in sampling the smooth function, sorted by the frequency based measure.

Population size	No. of Offsprings	No. of Bins	Neighborhood	Prob.of Survival	CC with Freq. based Measure	CC with Ent. based Measure
144.00	6.00	16.00	0.10	1.00	0.62	0.89
144.00	1.00	16.00	0.08	1.00	0.62	0.94
144.00	2.00	16.00	0.10	1.00	0.62	0.89
1024.00	6.00	16.00	0.05	1.00	0.62	0.94
64.00	1.00	16.00	0.10	1.00	0.61	0.92
144.00	2.00	16.00	0.10	0.90	0.61	0.89
64.00	1.00	16.00	0.05	0.90	0.61	0.94
144.00	1.00	16.00	0.10	1.00	0.61	0.92
144.00	1.00	16.00	0.05	1.00	0.61	0.94
144.00	2.00	16.00	0.08	0.90	0.61	0.92
4.00	1.00	16.00	0.10	1.00	0.61	0.92
144.00	6.00	16.00	0.10	0.90	0.61	0.88
144.00	1.00	16.00	0.08	0.90	0.61	0.93
4.00	1.00	16.00	0.08	1.00	0.61	0.93
144.00	1.00	16.00	0.08	0.80	0.61	0.92
144.00	1.00	16.00	0.05	0.90	0.61	0.94
1024.00	6.00	16.00	0.08	0.80	0.61	0.92
64.00	1.00	16.00	0.08	1.00	0.61	0.94
144.00	4.00	16.00	0.10	1.00	0.61	0.88
144.00	2.00	16.00	0.10	0.80	0.61	0.89
:	:	:	:	:	:	
:	:	:	:	:	:	
:	:	:	:	:	:	
144.00	6.00	4.00	0.05	0.80	0.15	0.73
64.00	6.00	4.00	0.08	0.80	0.15	0.71
64.00	4.00	4.00	0.05	0.70	0.14	0.69
4.00	2.00	4.00	0.08	0.90	0.14	0.70
144.00	2.00	4.00	0.08	0.80	0.14	0.69
64.00	6.00	4.00	0.05	0.70	0.14	0.67
4.00	1.00	4.00	0.05	0.90	0.14	0.71
64.00	2.00	4.00	0.08	1.00	0.14	0.72
64.00	2.00	4.00	0.08	0.80	0.13	0.69
144.00	6.00	4.00	0.08	0.80	0.13	0.71
64.00	2.00	4.00	0.08	0.90	0.13	0.70
144.00	2.00	4.00	0.05	0.80	0.12	0.74
144.00	2.00	4.00	0.08	0.90	0.12	0.69
144.00	2.00	4.00	0.08	1.00	0.11	0.70
144.00	1.00	4.00	0.05	1.00	0.10	0.69
144.00	2.00	4.00	0.05	0.70	0.09	0.71
64.00	1.00	4.00	0.05	1.00	0.08	0.67
144.00	4.00	4.00	0.05	0.70	0.08	0.70
144.00	6.00	4.00	0.05	0.70	0.07	0.68
4.00	1.00	4.00	0.05	1.00	0.05	0.67

Table 4.10: Performance of the Evolutionary Algorithm model in sampling the smooth function, sorted by the Entropy based measure.

Population size	No. of Offsprings	No. of Bins	Neighborhood	Prob.of Survival	CC with Freq. based Measure	CC with Ent. based Measure
1024.00	2.00	16.00	0.05	1.00	0.60	0.95
64.00	1.00	16.00	0.05	1.00	0.60	0.94
4.00	1.00	16.00	0.05	1.00	0.59	0.94
1024.00	4.00	16.00	0.08	1.00	0.60	0.94
1024.00	6.00	16.00	0.08	1.00	0.60	0.94
144.00	1.00	16.00	0.05	1.00	0.61	0.94
1024.00	2.00	16.00	0.08	1.00	0.60	0.94
144.00	1.00	16.00	0.05	0.90	0.61	0.94
1024.00	4.00	16.00	0.05	1.00	0.60	0.94
64.00	1.00	16.00	0.05	0.90	0.61	0.94
4.00	1.00	16.00	0.05	0.90	0.60	0.94
144.00	1.00	16.00	0.08	1.00	0.62	0.94
1024.00	4.00	16.00	0.02	1.00	0.57	0.94
1024.00	6.00	16.00	0.05	1.00	0.62	0.94
1024.00	6.00	16.00	0.08	0.90	0.59	0.94
64.00	1.00	16.00	0.08	0.90	0.60	0.94
1024.00	2.00	16.00	0.05	0.90	0.60	0.94
1024.00	4.00	32.00	0.08	1.00	0.55	0.94
4.00	1.00	16.00	0.05	0.80	0.60	0.94
64.00	1.00	16.00	0.08	1.00	0.61	0.94
:	:	:	:	:	:	
:	:	:	:	:	:	
:	:	:	:	:	:	
64.00	2.00	256.00	0.02	0.70	0.38	0.68
64.00	6.00	32.00	0.02	0.70	0.35	0.67
64.00	6.00	4.00	0.05	0.70	0.14	0.67
64.00	1.00	4.00	0.05	1.00	0.08	0.67
64.00	6.00	4.00	0.02	0.80	0.38	0.67
64.00	4.00	16.00	0.02	0.70	0.32	0.67
4.00	1.00	4.00	0.05	1.00	0.05	0.67
64.00	6.00	256.00	0.02	0.90	0.37	0.66
1024.00	1.00	4.00	0.10	0.80	0.29	0.66
64.00	4.00	256.00	0.02	0.80	0.33	0.66
64.00	4.00	4.00	0.02	0.70	0.30	0.66
64.00	6.00	16.00	0.02	0.70	0.33	0.65
64.00	6.00	256.00	0.02	0.80	0.41	0.65
1024.00	1.00	4.00	0.05	0.70	0.24	0.64
1024.00	1.00	4.00	0.08	0.70	0.27	0.64
64.00	4.00	256.00	0.02	0.70	0.41	0.63
1024.00	1.00	4.00	0.02	0.70	0.20	0.63
64.00	6.00	4.00	0.02	0.70	0.27	0.62
64.00	6.00	256.00	0.02	0.70	0.37	0.61
1024.00	1.00	4.00	0.10	0.70	0.23	0.57

Table 4.11: Performance of the Evolutionary Algorithm model in sampling the rapidly varying function, sorted by the frequency based measure.

Population size	No. of Offsprings	No. of Bins	Neighborhood	Prob.of Survival	CC with Freq. based Measure	CC with Ent. based Measure
4.00	2.00	16.00	0.05	1.00	0.87	0.83
4.00	2.00	32.00	0.05	1.00	0.87	0.83
64.00	2.00	256.00	0.05	1.00	0.87	0.82
4.00	2.00	256.00	0.05	1.00	0.87	0.84
64.00	2.00	16.00	0.05	1.00	0.87	0.79
64.00	6.00	256.00	0.05	1.00	0.87	0.81
4.00	4.00	16.00	0.05	1.00	0.87	0.82
4.00	6.00	16.00	0.05	1.00	0.87	0.82
64.00	2.00	16.00	0.05	0.80	0.87	0.81
64.00	4.00	256.00	0.05	1.00	0.87	0.82
4.00	4.00	32.00	0.05	1.00	0.86	0.83
64.00	2.00	32.00	0.05	1.00	0.86	0.79
64.00	4.00	16.00	0.05	1.00	0.86	0.78
64.00	4.00	32.00	0.05	0.90	0.86	0.80
64.00	6.00	32.00	0.05	1.00	0.86	0.80
64.00	2.00	16.00	0.05	0.90	0.86	0.80
4.00	6.00	16.00	0.05	0.90	0.86	0.85
64.00	6.00	16.00	0.05	1.00	0.86	0.78
4.00	4.00	32.00	0.05	0.90	0.86	0.86
64.00	4.00	32.00	0.05	1.00	0.86	0.79
:	:	:	:	:	:	
:	:	:	:	:	:	
:	:	:	:	:	:	
4.00	2.00	4.00	0.10	0.80	0.36	0.64
4.00	1.00	4.00	0.05	1.00	0.36	0.65
4.00	1.00	4.00	0.10	1.00	0.36	0.65
144.00	1.00	4.00	0.08	0.90	0.35	0.66
64.00	1.00	4.00	0.02	0.70	0.35	0.66
64.00	1.00	4.00	0.05	0.80	0.35	0.65
64.00	1.00	4.00	0.08	0.90	0.35	0.65
64.00	1.00	4.00	0.02	0.80	0.34	0.65
4.00	1.00	4.00	0.02	0.70	0.34	0.65
144.00	1.00	4.00	0.05	0.80	0.34	0.66
144.00	1.00	4.00	0.02	0.80	0.34	0.65
1024.00	1.00	4.00	0.08	0.70	0.34	0.57
4.00	1.00	4.00	0.02	0.80	0.34	0.65
4.00	1.00	4.00	0.05	0.80	0.33	0.64
144.00	1.00	4.00	0.08	1.00	0.33	0.63
144.00	1.00	4.00	0.05	0.90	0.32	0.63
64.00	1.00	4.00	0.05	0.90	0.32	0.64
64.00	1.00	4.00	0.08	1.00	0.32	0.63
4.00	1.00	4.00	0.08	1.00	0.32	0.63
4.00	1.00	4.00	0.05	0.90	0.32	0.63

Table 4.12: Performance of the Evolutionary Algorithm model in sampling the rapidly varying function, sorted by the entropy based measure.

Population size	No. of Offsprings	No. of Bins	Neighborhood	Prob.of Survival	CC with Freq. based Measure	CC with Ent. based Measure
4.00	1.00	32.00	0.02	1.00	0.76	0.93
64.00	1.00	32.00	0.02	1.00	0.74	0.93
4.00	1.00	256.00	0.02	1.00	0.77	0.92
64.00	1.00	256.00	0.02	1.00	0.72	0.91
144.00	1.00	32.00	0.02	1.00	0.67	0.90
4.00	2.00	32.00	0.05	0.80	0.81	0.89
64.00	1.00	16.00	0.02	1.00	0.72	0.89
4.00	1.00	16.00	0.02	1.00	0.73	0.88
4.00	2.00	256.00	0.05	0.90	0.84	0.88
4.00	6.00	32.00	0.05	0.80	0.82	0.88
4.00	4.00	32.00	0.05	0.80	0.82	0.88
4.00	2.00	256.00	0.05	0.80	0.74	0.88
4.00	6.00	32.00	0.08	1.00	0.79	0.88
144.00	2.00	256.00	0.05	1.00	0.82	0.88
144.00	1.00	16.00	0.02	1.00	0.67	0.88
4.00	6.00	256.00	0.05	0.90	0.83	0.87
4.00	4.00	32.00	0.08	1.00	0.79	0.87
4.00	4.00	32.00	0.05	0.70	0.74	0.87
4.00	2.00	16.00	0.05	0.80	0.81	0.87
4.00	4.00	256.00	0.05	0.90	0.83	0.87
:	:	:	:	:	:	
:	:	:	:	:	:	
:	:	:	:	:	:	
1024.00	1.00	16.00	0.08	0.70	0.41	0.65
1024.00	1.00	4.00	0.05	0.80	0.38	0.65
4.00	1.00	4.00	0.05	0.80	0.33	0.64
4.00	2.00	4.00	0.10	0.80	0.36	0.64
64.00	1.00	4.00	0.05	0.90	0.32	0.64
144.00	1.00	4.00	0.08	1.00	0.33	0.63
1024.00	1.00	4.00	0.02	0.70	0.38	0.63
64.00	1.00	4.00	0.08	1.00	0.32	0.63
64.00	6.00	4.00	0.02	0.80	0.61	0.63
4.00	1.00	4.00	0.08	1.00	0.32	0.63
1024.00	1.00	256.00	0.10	0.70	0.44	0.63
144.00	1.00	4.00	0.05	0.90	0.32	0.63
64.00	4.00	4.00	0.02	0.70	0.60	0.63
4.00	1.00	4.00	0.05	0.90	0.32	0.63
64.00	6.00	4.00	0.02	0.70	0.60	0.62
1024.00	1.00	4.00	0.05	0.70	0.36	0.62
1024.00	1.00	4.00	0.10	0.80	0.37	0.61
1024.00	1.00	16.00	0.10	0.70	0.40	0.61
1024.00	1.00	4.00	0.10	0.70	0.38	0.58
1024.00	1.00	4.00	0.08	0.70	0.34	0.57

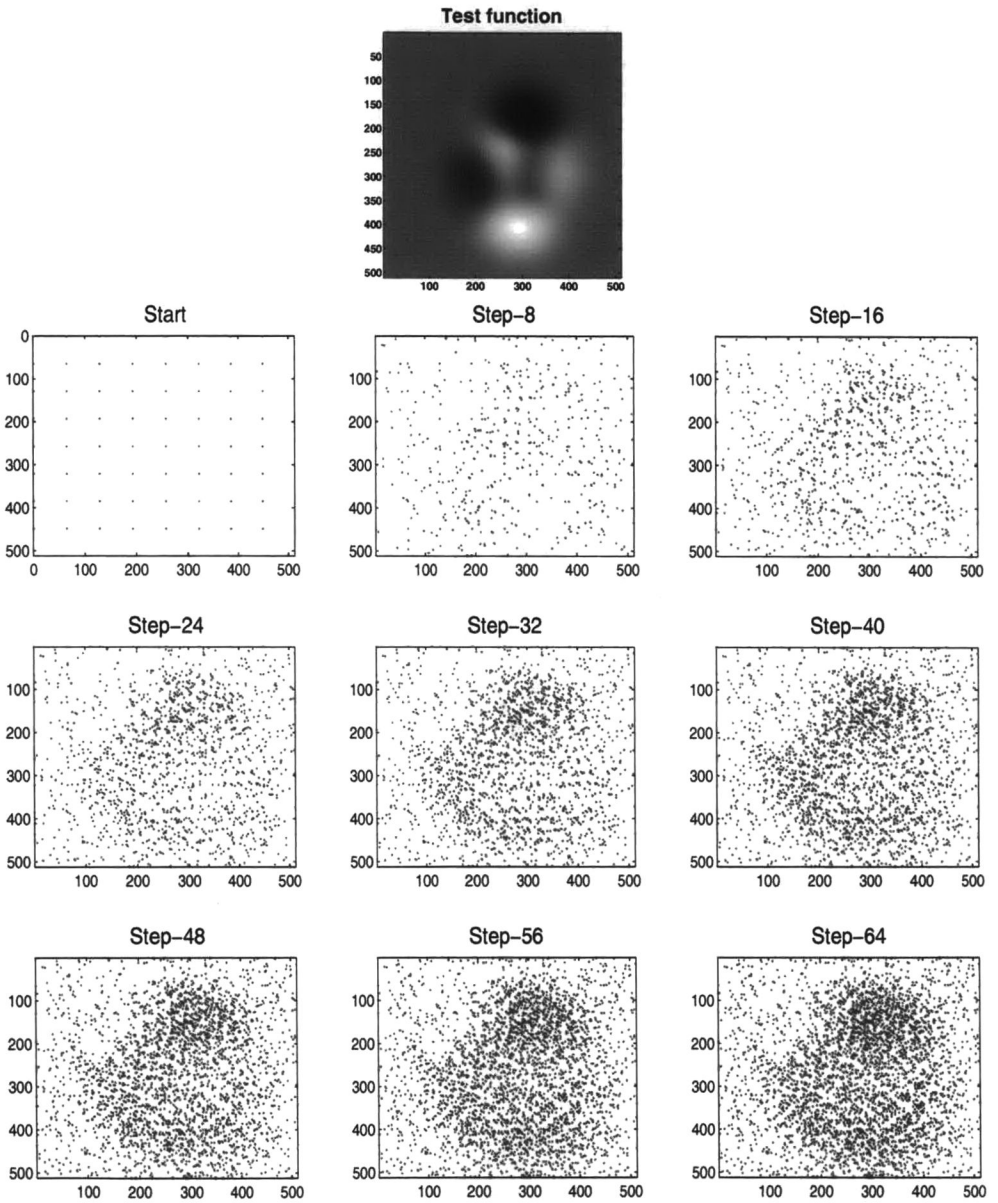


Figure 4.19: Intermediate stages from sampling a smooth test function with the Evolutionary Algorithm model. The good performance is recorded by using the following factors: SOP = 64, N_{of} = 1, N_{ub} = 16, DOP = 0.05 and POS = 1.0. Performance measures = 0.60/0.94. Total of 64 generations.

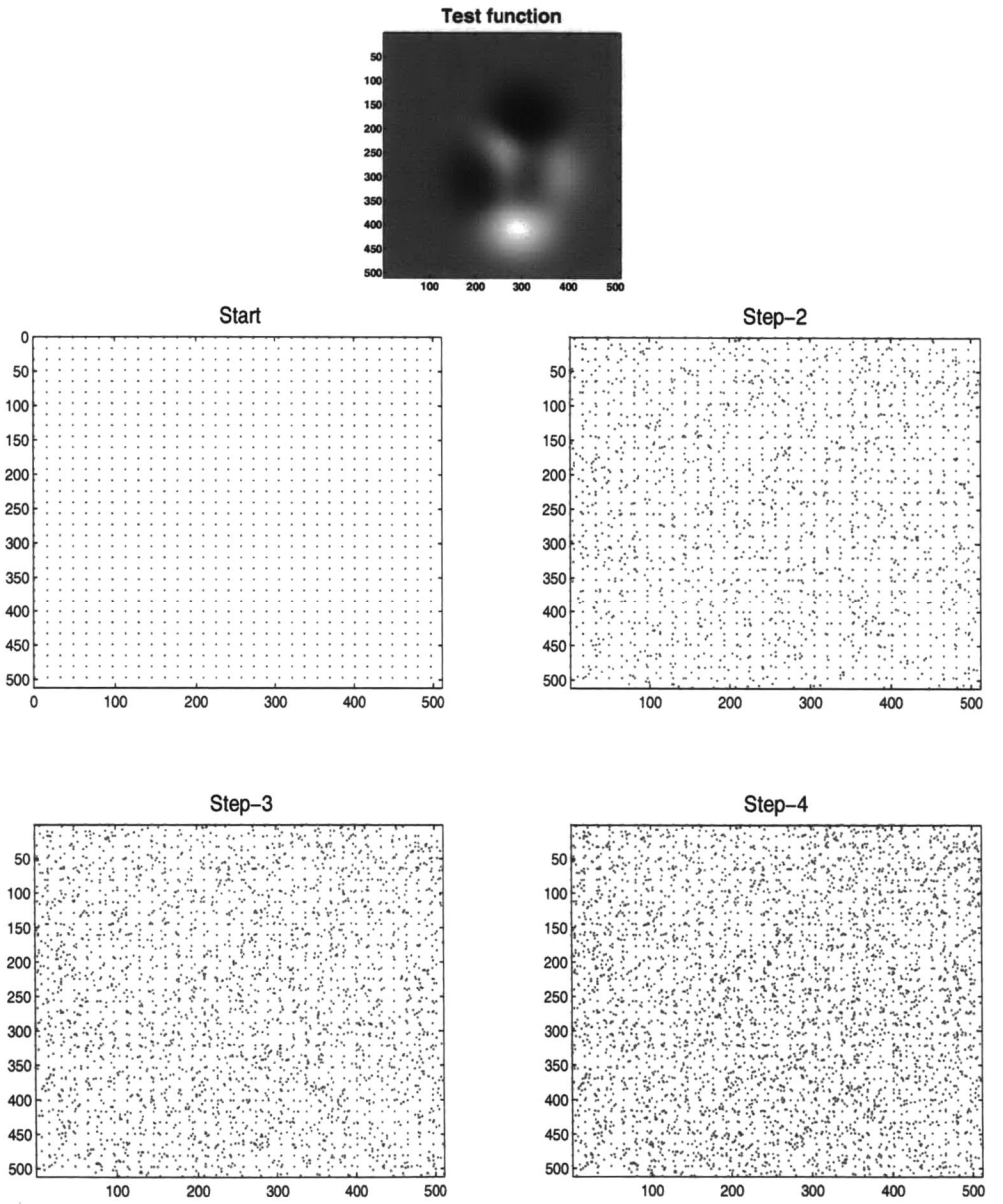


Figure 4.20: All the stages from sampling a smooth test function with the Evolutionary Algorithm model. The poor performance is recorded by using the following factors: SOP = 1,024, N_{of} = 1, N_{ub} = 4, DOP = 0.10 and POS = 0.7. Performance measures = 0.23/0.57. Total of four generations.

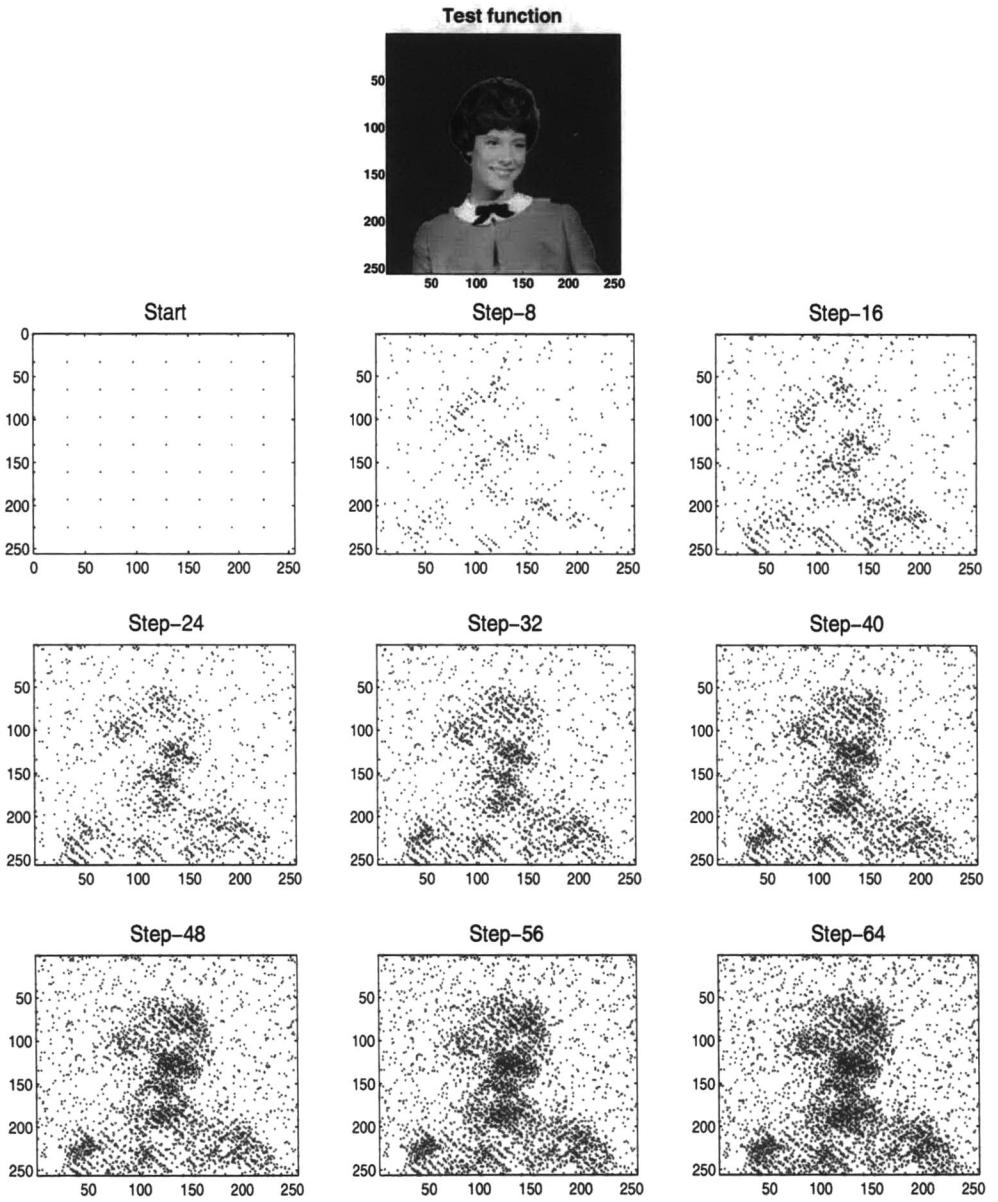


Figure 4.21: Intermediate stages from sampling a rapidly varying test function with the Evolutionary Algorithm model. The good performance is recorded by using the following factors: $SOP = 64$, $N_{\text{of}} = 1$, $N_{\text{ub}} = 32$, $DOP = 0.02$ and $POS = 1.0$. Performance measures = 0.76/0.93. Total of 64 generations.

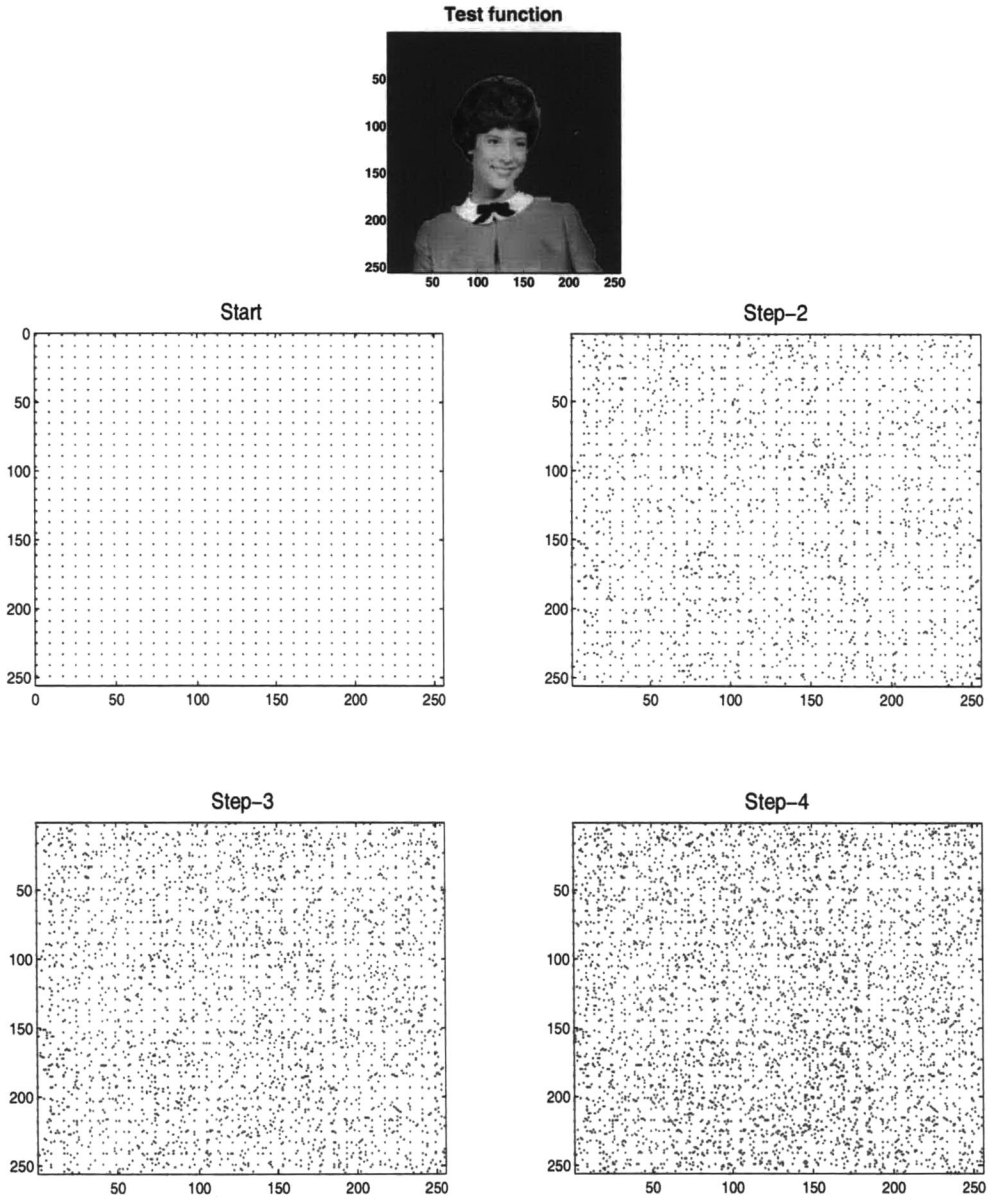


Figure 4.22: All the stages from sampling a rapidly varying test function with the Evolutionary Algorithm model. The poor performance is recorded by using the following factors: SOP = 1,024, N_{of} = 1, N_{ub} = 4, DOP = 0.08 and POS = 0.7. Performance measures = 0.23/0.57. Total of four generations.

The performance of the three models is summarized in Tables 4.13(a) - (c). They show the ranges of factors that results in the good performance recorded for each model. The tables also show corresponding averages of the entropy based performance measure listed in Tables 4.2, 4.4, 4.6, 4.8, 4.10, and 4.12.

Table 4.13: A summary of the results from the analysis of the three sampling models.

(a) Summary results for the active walker model.

	No. of active walkers	No. of bins	Long step	Short step	Average CC
Smooth	4 - 144	16 - 32	0.35 - 0.4	0.04 - 0.08	0.92
Rapidly varying	4 - 144	16 - 32	0.30 - 0.4	0.02	0.85

(b) Summary results for the ant model.

	No. of ants	No. of bins	Range of pheromone	Range of foraging	Average CC
Smooth	144 - 256	8 - 16	0.02 - 0.25	0.10 - 0.15	0.93
Rapidly varying	144 - 256	8 - 32	0.02 - 0.25	0.10 - 0.12	0.90

(c) Summary result for the evolutionary algorithm model.

	Pop. size	No. of offspring	No. of bins	Neighb-orhood	Prob. of survival	Average CC
Smooth	4 - 1024	1 - 6	16 - 32	0.02 - 0.08	0.8 - 1.0	0.94
Rapidly varying	4 - 144	1 - 6	16 - 256	0.02 - 0.08	0.7 - 1.0	0.89

The table shows the ranges of input factors that result in good performance, and averages of the entropy based performance values. The factors in the tables are described in Sections 4.3, 4.4, and 4.5. The shown average correlation coefficient (CC) values are the means of the entropy based performance values, which is computed as the CC between the sample density and the entropy in a region of the sample space.

4.6 Further Analysis of the Active Walker Model

Based on our discussions so far, the active walker model is the most useful of the three considered. This is because of our ability to correlate each factor in the model to its performance. The active walker model is thus given further consideration by investigating its scaling properties. We extend our analysis to three dimensions by considering a 3-dimensional test function defined as $(Sinc(x) \times Sinc(y) \times Sinc(z))$, of size $512 \times 512 \times 20$. Figure 4.23 shows a 2-dimensional slice of the function.

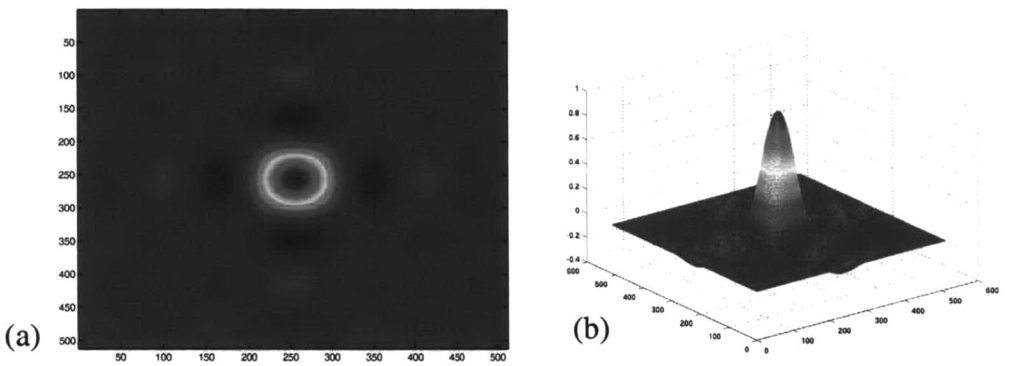


Figure 4.23: *Slice* from 3-dimensional Sinc test function. (a) top view, (b) side view.

We obtain $81,920$ samples, resulting in the same sample ratio of $1 : 64$ used in the test of the 2-dimensional functions considered earlier. Our derived measures of objective performance are extended for the 3-dimensional case. Our subsequent analysis are similar to that done for the 2-dimensional functions. Tables 4.14 and 4.15 show results of our tests, sorted in a manner similar to that done in the 2-dimensional analysis. The number of active walkers was scaled by 20, the size of the third dimension. This is to ensure proper comparison to the 2-dimensional experiment, especially on the basis of the number of steps taken by the active walkers. Note the similarities between the factors that result in good performance for the 2-, and 3-dimensional functions. This is an indi-

cation that the performance of the active walker model does not change appreciably with change in number of dimensions.

4.7 Conclusions

The findings of the analysis of the three models employed in implementing the ASHE algorithm are summarized as follows:

1. Generally, the entropy measure indicates that all three models performed better than the frequency based measure. This is indicated by the higher positive CC values. The frequency based measure is limited by the small spatial sample used in the experiments. This results in poor frequency resolution.

2. Using both measures of performance, the EA, and ant models performed marginally better than the active walker model.

3. The ant, and EA models show no apparent correlation between one or more factors, and their performance. This makes it difficult to come up with a combination of factors that are appropriate for a particular application.

4. We are able to establish correlations, separately, between the active walker model factors, and the performance of the model. This makes it possible to establish general "rules of thumb" in its application.

5. The active walker model is more robust since there is always a possibility for walkers to sample in all regions of the space all through the sampling process. In the process of sampling using the ant and EA models, some regions may be completely excluded due to good solutions obtained from other regions. This is similar to obtaining a local minimum.

Table 4.14: Performance of Active Walker model in sampling the 3-dimensional Sinc function, sorted by the frequency based measure.

No. of Walkers	No. of Bins	Long step	Short step	CC with Freq. based Measure	CC with Ent. based Measure
1280.00	8.00	0.40	0.04	0.97	0.97
2000.00	8.00	0.35	0.04	0.97	0.97
2000.00	8.00	0.40	0.04	0.97	0.97
2880.00	8.00	0.40	0.04	0.97	0.97
1280.00	8.00	0.35	0.04	0.97	0.97
2880.00	8.00	0.35	0.04	0.97	0.97
2880.00	8.00	0.30	0.04	0.97	0.96
80.00	8.00	0.40	0.04	0.96	0.96
80.00	8.00	0.35	0.04	0.96	0.96
1280.00	8.00	0.30	0.04	0.96	0.96
80.00	8.00	0.30	0.04	0.96	0.96
2000.00	8.00	0.30	0.04	0.96	0.96
80.00	8.00	0.40	0.06	0.96	0.95
2000.00	8.00	0.25	0.04	0.96	0.95
2880.00	8.00	0.25	0.04	0.96	0.95
1280.00	8.00	0.25	0.04	0.96	0.95
80.00	8.00	0.35	0.06	0.95	0.95
1280.00	8.00	0.40	0.06	0.95	0.95
1280.00	8.00	0.35	0.06	0.95	0.95
2880.00	8.00	0.40	0.06	0.95	0.95
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
2000.00	32.00	0.25	0.10	-0.14	-0.15
20480.00	8.00	0.20	0.10	-0.20	-0.20
2000.00	256.00	0.20	0.10	-0.21	-0.20
2880.00	256.00	0.20	0.10	-0.24	-0.23
1280.00	256.00	0.20	0.10	-0.26	-0.25
80.00	256.00	0.20	0.10	-0.28	-0.27
20480.00	16.00	0.20	0.10	-0.31	-0.31
20480.00	32.00	0.20	0.10	-0.36	-0.36
2000.00	8.00	0.20	0.10	-0.44	-0.45
2880.00	8.00	0.20	0.10	-0.46	-0.46
1280.00	8.00	0.20	0.10	-0.47	-0.47
80.00	8.00	0.20	0.10	-0.49	-0.49
2880.00	16.00	0.20	0.10	-0.60	-0.60
2000.00	16.00	0.20	0.10	-0.60	-0.60
1280.00	32.00	0.20	0.10	-0.61	-0.60
2880.00	32.00	0.20	0.10	-0.62	-0.61
1280.00	16.00	0.20	0.10	-0.63	-0.62
80.00	32.00	0.20	0.10	-0.63	-0.62
2000.00	32.00	0.20	0.10	-0.64	-0.62
80.00	16.00	0.20	0.10	-0.65	-0.64

Table 4.15: Performance of Active Walker model in sampling the 3-dimensional Sinc function, sorted by the entropy based measure.

No. of Walkers	No. of Bins	Long step	Short step	CC with Freq. based Measure	CC with Ent. based Measure
1280.00	8.00	0.40	0.04	0.97	0.97
2000.00	8.00	0.35	0.04	0.97	0.97
2000.00	8.00	0.40	0.04	0.97	0.97
2880.00	8.00	0.40	0.04	0.97	0.97
1280.00	8.00	0.35	0.04	0.97	0.97
2880.00	8.00	0.35	0.04	0.97	0.97
2880.00	8.00	0.30	0.04	0.97	0.96
80.00	8.00	0.40	0.04	0.96	0.96
80.00	8.00	0.35	0.04	0.96	0.96
1280.00	8.00	0.30	0.04	0.96	0.96
80.00	8.00	0.30	0.04	0.96	0.96
2000.00	8.00	0.30	0.04	0.96	0.96
2000.00	8.00	0.25	0.04	0.96	0.95
2880.00	8.00	0.25	0.04	0.96	0.95
1280.00	8.00	0.25	0.04	0.96	0.95
80.00	8.00	0.40	0.06	0.96	0.95
80.00	8.00	0.35	0.06	0.95	0.95
1280.00	8.00	0.40	0.06	0.95	0.95
2000.00	8.00	0.40	0.06	0.95	0.95
1280.00	8.00	0.35	0.06	0.95	0.95
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
2000.00	32.00	0.25	0.10	-0.14	-0.15
2000.00	256.00	0.20	0.10	-0.21	-0.20
20480.00	8.00	0.20	0.10	-0.20	-0.20
2880.00	256.00	0.20	0.10	-0.24	-0.23
1280.00	256.00	0.20	0.10	-0.26	-0.25
80.00	256.00	0.20	0.10	-0.28	-0.27
20480.00	16.00	0.20	0.10	-0.31	-0.31
20480.00	32.00	0.20	0.10	-0.36	-0.36
2000.00	8.00	0.20	0.10	-0.44	-0.45
2880.00	8.00	0.20	0.10	-0.46	-0.46
1280.00	8.00	0.20	0.10	-0.47	-0.47
80.00	8.00	0.20	0.10	-0.49	-0.49
2000.00	16.00	0.20	0.10	-0.60	-0.60
2880.00	16.00	0.20	0.10	-0.60	-0.60
1280.00	32.00	0.20	0.10	-0.61	-0.60
2880.00	32.00	0.20	0.10	-0.62	-0.61
1280.00	16.00	0.20	0.10	-0.63	-0.62
80.00	32.00	0.20	0.10	-0.63	-0.62
2000.00	32.00	0.20	0.10	-0.64	-0.62
80.00	16.00	0.20	0.10	-0.65	-0.64

6. Further experiments with the active walker model indicate that it scales well.

In addition to the foregoing, it is also straightforward to extend the active walker model into sampling in higher dimensions. The locations in an n-dimensional space are defined as vectors of length n, and distances moved by the active walkers are computed as simple vector operations. These are the reasons for the choice of the active walker model for the application discussed in Chapter 6.

CHAPTER 5

NATURE, USES AND SYNTHESIS OF HYPERSPECTRAL IMAGES

Here, we give a description of the nature, and uses of hyperspectral images. Next, we state the need for image synthesis, and describe the process in general. Finally, the process of generating a database of hyperspectral images is described, and some of the results are shown.

5.1 Nature of Hyperspectral Images

Hyperspectral images are *cubes* of data, with each value in the *cube* representing the electromagnetic energy response from an imaged scene, at a particular wavelength. Two of the dimensions in the *cube* are spatial, and the third dimension is spectral. That is, each spectral component, called a *band*, is made up of a 2-dimensional spatial image. The bands in a hyperspectral image are contiguous, and occupy a region of the electromagnetic spectrum. For example, an image with contiguous spectral bands of wavelengths in the micrometer range will be a hyperspectral infrared (HSI) image because of its location on the electromagnetic spectrum. Assuming that the bands are not completely correlated, integration of data in more than one band will result in increase in information about an image. There is usually a level of independence between the bands, and this results in a spectral signature for each spatial pixel. That is, an imaging device will record varying responses at the different wavelengths in the hyperspectral image. These responses depend on the intrinsic nature of the imaged material, thus a unique signature is recorded for each material. The information in the spectral signature is particularly useful for, but not limited to situations, in which there is a limitation on the spatial resolution that can

be obtained. Figure 5.1 shows an example of a hyperspectral image, and illustrates the foregoing about their nature.

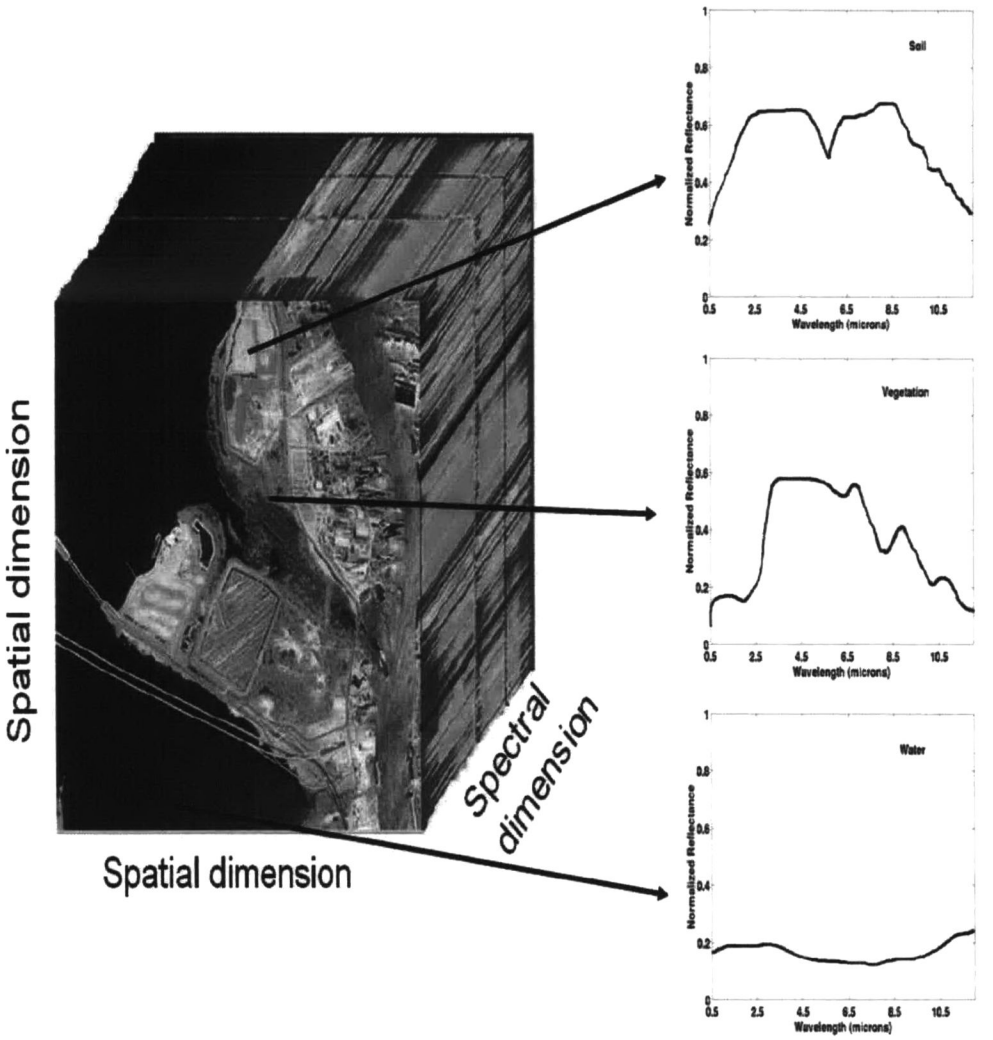


Figure 5.1: Example hyperspectral image, and material signatures. (Source: IEEE Signal Processing Magazine, Vol.19(1), 2002.)

Using multi-spectral Automatic Target Recognition (ATR) algorithms, objects in hyperspectral scenes that only span one pixel in the spatial dimensions or are even sub-pixel, can be identified from their spectral signatures. Generally, the approach of multi-spectral

ATR algorithms focuses on the spectral rather than the spatial information in the images [50]. There are numerous military [7, 44], and non-military [89, 90, 5, 93, 4] uses of multi-, and hyperspectral images.

5.2 Synthesizing Hyperspectral Images

Hyperspectral images obtained for military purposes are generally not available in the public domain because of security reasons. Even when images are available, they usually do not exist in the quantity or with the specifications required by many applications. A solution to this problem is to synthesize images with these required specifications. Synthetic images have been used as aid in the design and development stages of imaging sensors by providing an avenue to pre-evaluate the imaging products from the sensors [54, 73]. They also serve as test data for algorithm design, either because of the lack of real data [1, 80], or to augment the available real data [77]. Some examples of Synthetic Image Generation (SIG) models are the Strategic High Altitude Atmospheric Radiance Code (SHARC) [9], full spectrum scene simulator (MCScene) [70], and the Digital Imaging and Remote Sensing Image Generation (DIRSIG) [76]. All these models generate scenes by tracing rays between a simulated imaged scene, and an imaging sensor. Models of the intervening space between these two are included in the ray-tracing process.

The images we require are used primarily in a military application described in Chapter 7. The DIRSIG model has been used extensively in military applications because of the good radiometric fidelity of the images it generates. We thus decided to use this model for image synthesis.

5.2.1 The Digital Imaging and Remote Sensing Image Generation model (DIRSIG)

The model is an integrated collection of first principle based sub-models that account for scene geometry, atmospheric contributions, illuminating sources, and properties of materials in the imaged scene. After these factors are established, a ray-tracing process is employed in rendering the scene. It has be been used for generating high spatial and spectral, multi- or hyperspectral images in the 0.3 to 20 micron region [76]. The following is a brief description of some of the components of, and tools used by DIRSIG. A full description can be found in the DIRSIG manual [10].

Scene

This is a 3-dimensional space, and comprises of terrains and objects. Each of these consists of single or multiple facets. Associated with each facet in a scene are pre-defined radiometric properties obtained from experimenting with different materials. These properties determine the response from the surfaces as recorded by the imaging sensor. The shape, and number of facets on an object is fixed, but the user is allowed to associate any material with a facet. The user is also allowed to define the 3-dimensional location, size, and orientation of objects in the scene. For the imaging geometry, the relative positions of the sensors and scene can be specified in the 3-dimensional coordinate system, or in terms of distances from sensor to scene and angles relative to a reference. This allows for all the practical imaging geometries that may be needed.

Sensors

All the sensors modeled in DIRSIG are passive. This means that they register the energy that is reflected from an external source, or the energy that is emitted from the object itself. Some examples are frame cameras, and line scanners. Example parameters that

may be set for these sensors are focal length, flight paths (for sensors mounted on moving carriers), number of scan lines, and number of samples per line.

Radiometry

DIRSIG uses the MODerate spectral resolution atmospheric TRANsmittance (MODTRAN) algorithm and computer model [6] for its radiometric computations. It utilizes bidirectional reflectance data, and accounts for specular and diffuse background contributions. It also models length dependent extinction and emission properties of plumes, clouds, targets, and backgrounds. In summary, it models the intervening space between an imaged scene and a sensor. Based on this model, a database or lookup table of values is computed for each pixel in every spectral band. MODTRAN has a current limitation of 2cm^{-1} spectral resolution.

Ray Tracing

A ray tracing component utilizes the geometry information to generate a list of facets intersecting a given pixel. This is combined with information from the the radiometry model, and used in the radiance computations.

Other Software Tools

The DIRSIG comes packaged with an image viewing software called FREELOOK. This is used for previewing, and for spectral analysis of the generated hyperspectral images.

5.3 Image Synthesis with DIRSIG

The images we synthesized are used as aid in the development of Automatic Target Recognition (ATR) algorithms. Specifically, they serve as test images, used in the eval-

uation of the performance of ATR algorithms developed for military applications. The required images are Forward-Looking Infrared Images (FLIR). The database is generated according to the following specifications for each image:

1. Sensor type: single-shot images are required, thus a framing array sensor is used. This sensor's focal length is set to 50 mm.

2. Imaging geometry: the sensor is placed at a stand-off from the imaged scene in a forward-looking arrangement. The distance between the sensor and the imaged scene is 2 km. The sensor is elevated at 50 m above the imaged scene to give a larger field of view.

3. Spatial resolution: each band is of size 512×512 pixels. The spatial resolution is computed by using similar triangles. This is computed based on the DIRSIG default image length and breadth of 24748.7 microns, a framing array sensor of focal length 50 mm, and the distance between the sensor and imaged scene of 2 km. This results in a resolution of 1.93 meters.

4. Spectral span and resolution: the images range in wavelength from $8 - 13$ microns, with 40 nanometer steps between bands. This results in 126 bands per image.

Based on the stated use of the images, it is required that there is diversity in the database with respect to ATR performance. That is, images of varying degrees of difficulty should be represented in order that the ATR algorithms are adequately tested. We attempt to manually include such diversity, by varying the following factors in the imaged scene:

1. Time of day: we generate images for two different times of the day. These are 3.00 AM in the morning and 3.00 PM in the afternoon. Changes in this factor will

generally result in radiometric changes.

2. Clutter: all objects different from the target of interest are considered clutter. This includes all objects and background that can be mistaken for, or hinder in the detection of a target of interest. We introduce clutter of varying types, and in different quantities into the scenes. DIRSIG has models for both man-made clutter such as fuel drums and tents, and natural clutter such as trees and hilly terrains.

3. Target: we generate some images with a military truck as target, and others with an armored tank.

A combination of all these results in 216 hyperspectral images. Figure 5.2 shows the combination of factors that result in the database. The information containing all the image specifications are included in configuration text files required as arguments by DIRSIG for execution.

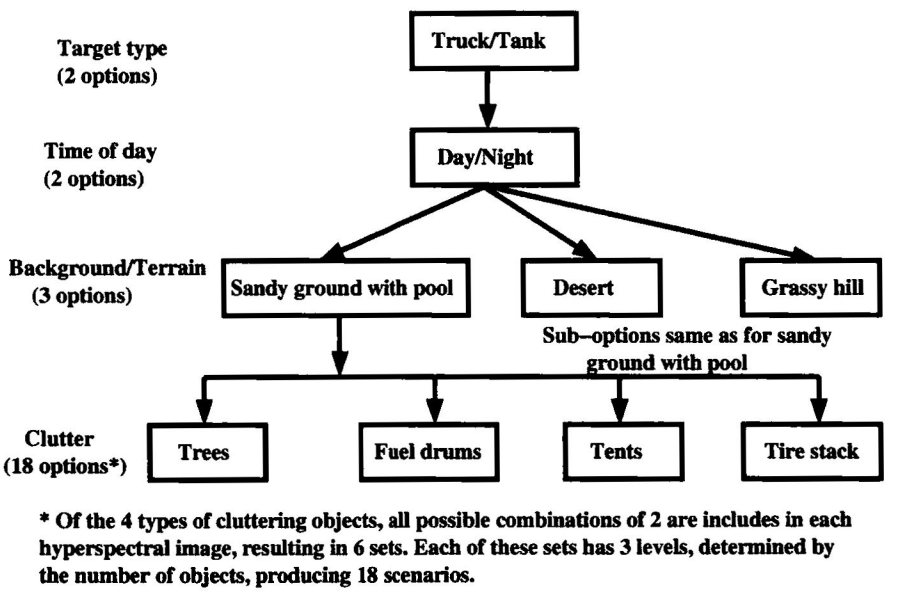


Figure 5.2: Combination of factors used to generate images in synthesized database. Total of 216 images.

Figures 5.3 - 5.5 show some example images from the synthesized database. Figure 5.6 shows example spectral signatures from some of the images.

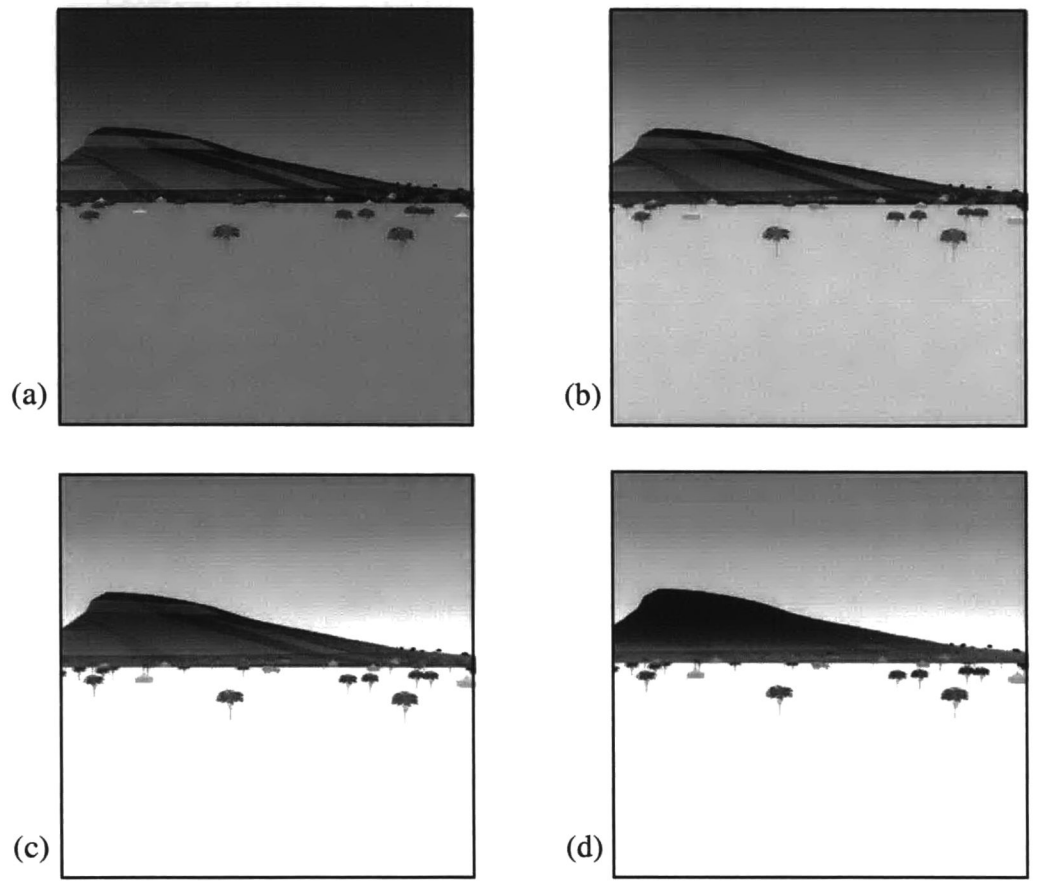


Figure 5.3: Hyperspectral image scene with a target truck on a flat surface with a hilly background. The cluttering objects are tents and trees. The wavelengths of the shown bands are (a) $\lambda = 8$ microns, (b) $\lambda = 10.6$ microns, (c) $\lambda = 11.96$ microns, (d) $\lambda = 12.76$ microns.

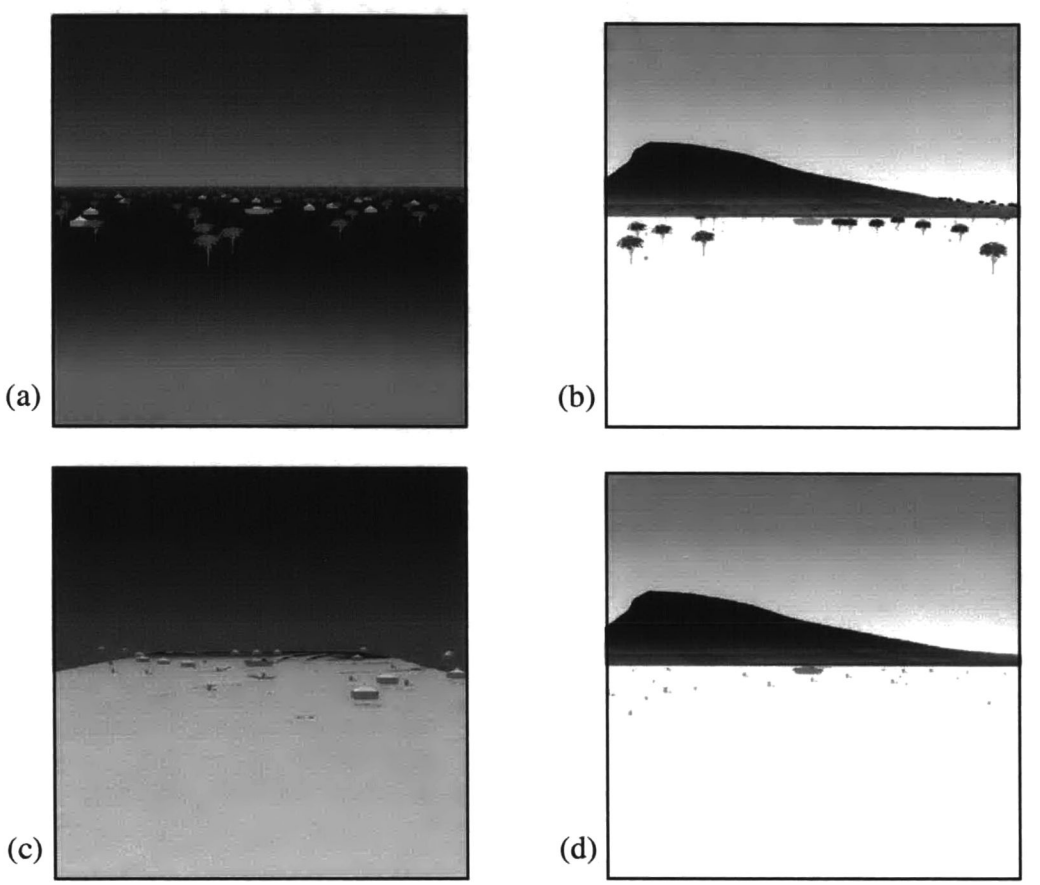


Figure 5.4: Images of different bands from different scenes, with the same target tank. (a) flat sand ground with trees and closed tents as cluttering objects, $\lambda = 8.36$ microns, (b) flat ground with hilly background, trees, and fuel drums as cluttering objects, $\lambda = 12.76$ microns, (c) desert terrain with closed tents and tire stacks as cluttering objects, $\lambda = 8.36$ microns, and (d) flat ground with hilly background, tire stacks and fuel drums as clutter, $\lambda = 12.76$ microns.

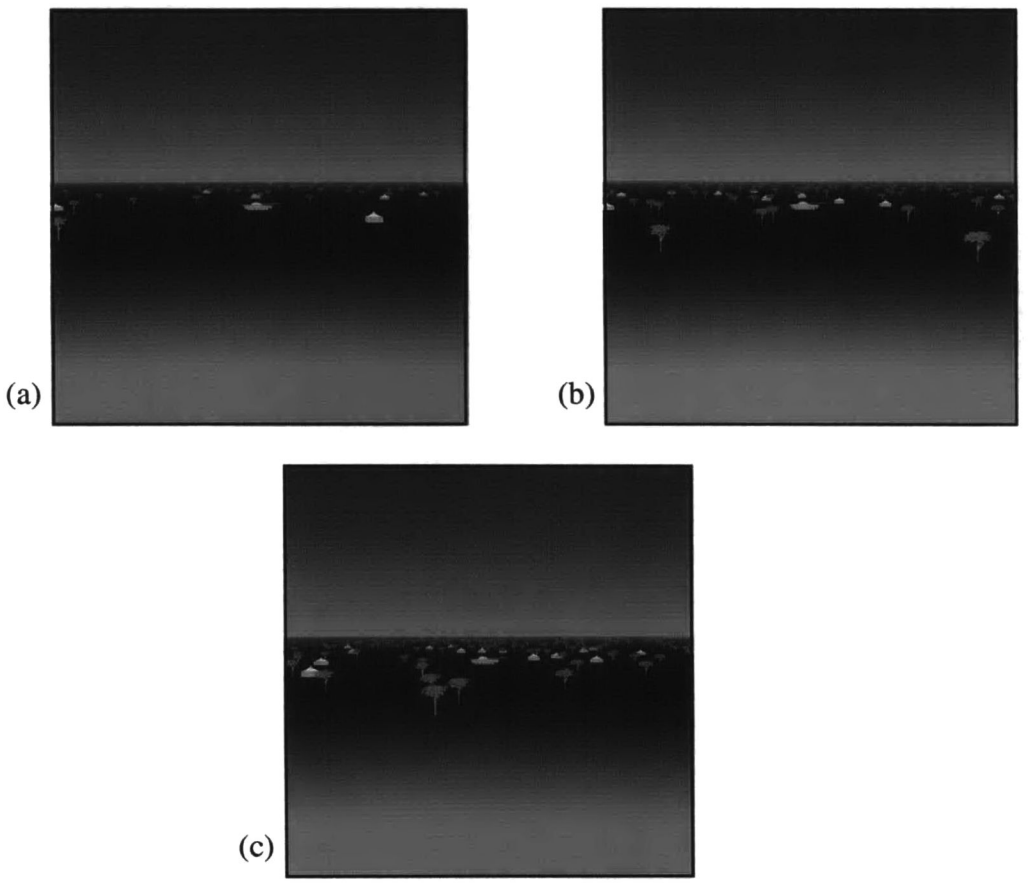


Figure 5.5: Example images of the same scene, with varying levels of clutter, determined by the quantity. (a) low clutter, (b) medium clutter, and (c) high clutter.

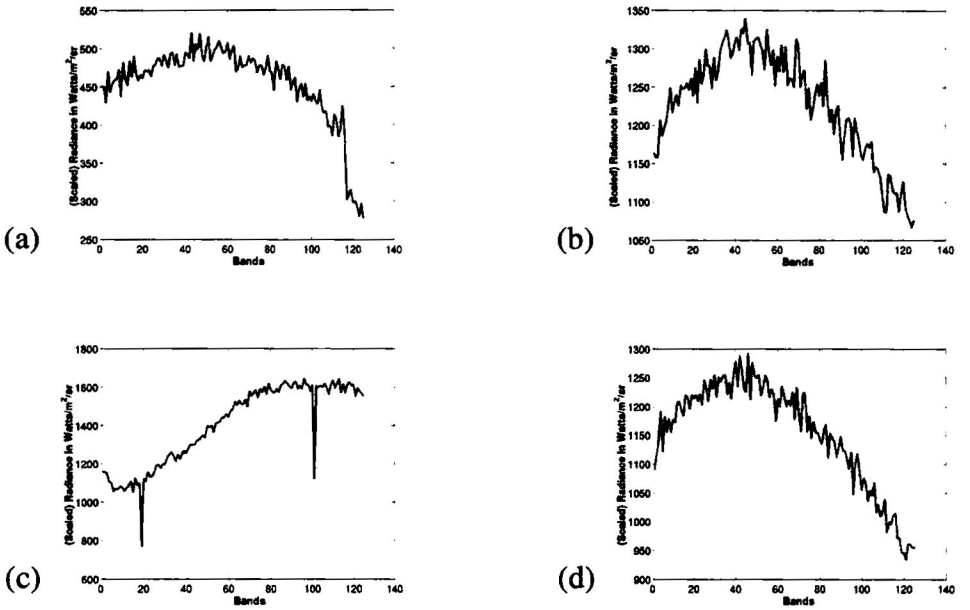


Figure 5.6: Examples of spectral signatures of different materials in the synthesized hyperspectral scenes. (a) side of hilly background, (b) side of closed tent, (c) flat sandy ground, and (d) metal front of truck.

In conclusion, it is important to note that the image synthesis process is computationally expensive. Each hyperspectral image in the database took about 150 minutes to synthesize on a 3.2 GHz Pentium IV processor machine, with 2 GB of memory. The whole database creation took about 540 hrs. DIRSIG stores each pixel as a float. Each synthesized hyperspectral image is of size 512×512 pixels $\times$ 126 bands $=$ 126MB.

CHAPTER 6

EFFICIENT HYPERSPECTRAL IMAGE SYNTHESIS USING ASHE

In this chapter, we present a more efficient approach to the hyperspectral image synthesis process described in Chapter 5. This approach is based on the Adaptive Sampling by Histogram Equalization (ASHE) algorithm. As mentioned in the hyperspectral image synthesis discussion, our aim is to generate a set of images utilized in the performance evaluation of Automatic Target Recognition (ATR) algorithms. In general, data analysis of any sort requires adequate, and a statistically representative population of the dataset in question in order to make reliable inferences. Two specific requirements of our synthesized images are:

1. Fidelity of each image. This depends on the ability of the synthetic image generation system to adequately model, and reproduce the complex interactions that exist in a real scene. There is continuous research work aimed at developing this ability [74]. It is however, beyond the scope of this work.

2. Representation in all categories of ATR difficulty in the database.

The latter requirement is the focus of this application, and it ensures that the ATRs in question are evaluated for all levels of target detection and recognition. This is an important requirement for drawing an unbiased, and conclusive inference about the performance of ATRs. The following sections describe the process of image synthesis based on the ASHE algorithm, and present some results.

6.1 ASHE based Image Synthesis

We model each generated image as a function of multiple factors, each image is thus a point in the multi-dimensional space. Some of these factors, such as time of day, are described in Section 5.3. Each synthesized image is thus a result of combining these factors as inputs to the DIRSIG model. Joining these points obtained from all possible combinations of factors yields a surface in the space. There is usually no prior knowledge of how a particular combination of conditions will affect the performance of an ATR. Without such knowledge, the typical approach is to generate images for a random combination of factors, or to generate images for combinations of factors that are evenly spaced within their possible ranges. These approaches are inefficient for situations in which there are varying slopes in the described multi-factor space. Also, due to the computational complexity of hyperspectral image synthesis described in Section 5.3, a *brute force* approach, which requires the generation of images from all combinations of factors is not feasible. Other approaches such as the gradient based search are also not feasible for the same reason.

The optimal reconstruction of such a surface from a limited number of points will result from concentrating relatively more points in regions of rapid image variation. Thus, it is desired to generate images for values, or ranges of values of these factors that are significant for change in target recognition difficulty. As shown in the description of the ASHE algorithm in Chapter 2, sampling this surface in this manner results in a distribution of ATR performance values that tends towards the uniform. Thus, sampling the surface to maximize diversity in values indicative of ATR performance, results in efficient sampling of the surface. We employ the ASHE algorithm, using the active walker model to sample the surface. The decision to use the active walker model is based on our

conclusions in Section 4.7. It is straightforward to extend the described 2-dimensional version of the model to higher dimensions. More importantly, we are able to establish appropriate input parameters

6.1.1 Imaged Scene

For this experiment, we generate images according to the urban scene from the DIRSIG manual [10]. A single band from this scene, spatial size 128 × 128 pixels, is shown in Figure 6.1.

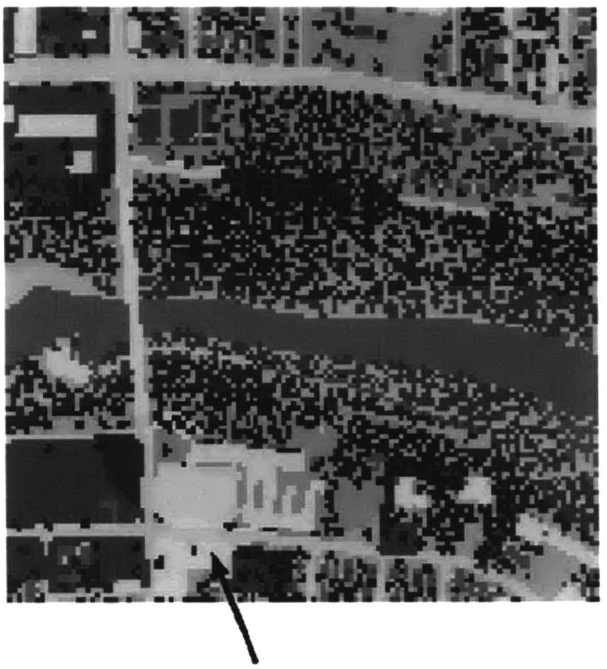


Figure 6.1: A single band ($\lambda = 0.56$ nm) from the hyperspectral image of the urban scene. The spatial size is 128×128 pixels. The arrow indicates the region cropped as target.

6.1.2 Input Factors to DIRSIG

We generate images from the visible to near infrared ($0.35 - 1.0$nm) regions of the electromagnetic spectrum. We identify factors that will generally result in radiometric changes, and thus spectral signatures, for this spectral range. Some of these are time of day, day of year, visibility parameter, aerosol type parameter, wind speed, and the parameter representing the modeled atmospheric profile. The significance of each of these factors is described in detail in the DIRSIG manual. We place a further constraint on the factors utilized in image synthesis. The extra requirement is that the image synthesis based on ASHE only utilizes factors that consist of ordered sets. This ensures that a move in any single dimension generally results in an increase or decrease in the radiometric effect from that factor. This criterion excludes the parameter for the aerosol type, and atmospheric profile. These are unordered sets, and the implication is that an active walker's movement in these dimensions is random. There has to be a correlation between the step sizes of the active walker in the input parameter space, and their sample contribution to the distribution. The wind speed factor was excluded based on further experience with the image synthesis process. Our image synthesis is thus based on the following three factors:

- Time of day ($1 - 24$ hours)
- Month of year ($1 - 12$)
- Visibility parameter ($0 - 40$ km)

6.1.3 Baseline ATR Performance

In order to utilize the ASHE algorithm in the image synthesis process, we need to associate a value, indicative of baseline ATR performance with each image. The ASHE

algorithm then attempts to equalize the distribution of these values, as the sampling process progresses. We establish this through the performance of an idealized ATR. We implemented a normalized, multi-spectral matched filter ATR via the Adaptive Coherence Estimator (ACE).

$$\mathbf{ACE_{statistic}} = \frac{|\mathbf{s}^T\hat{\mathbf{R}}_b^{-1}\mathbf{x}|^2}{(\mathbf{s}^T\hat{\mathbf{R}}_b^{-1}\mathbf{s})(\mathbf{x}^T\hat{\mathbf{R}}_b^{-1}\mathbf{x})} \tag{6.1}$$

The ATR uses a spectral signature of a target in question as a template. The resulting ACE statistic is bounded between 0 and 1, and it is expressed in (6.1), where $\mathbf{s} \in \mathbb{R}^L$ and $\mathbf{x} \in \mathbb{R}^L$ are the target template and pixel under test respectively, and L is the number of bands in the hyperspectral image. The vectors s and x may also be composed of multiple pixels in the spatial dimension. In this case, 2-dimensional averages of the target and test pixels are taken in the spatial dimensions to obtain column vectors of the previously stated lengths. $\hat{\mathbf{R}}_b$, with dimensions $L \times L$ is an estimate of the covariance matrix of the background [57].

This ATR is idealized since it uses a hyperspectral image target template that is cropped from the scene. A 3×3 pixel target is cropped from the area indicated by the arrow in Figure 6.1. A 2-dimensional average of this is taken in the spatial dimension to obtain a vector of length L = Number of bands, as described earlier. The false alarm rate at a particular threshold is an indication of the baseline ATR performance for a scene. The same threshold is used for all scenes to obtain this baseline performance. Note that this indicated performance is specific for the target. The use of different targets may result in a different false alarm rates at the same threshold. This is common practice, since most practical ATR algorithms are evaluated based on the detection of specific targets using the known target template. The diversity in the synthesized images is thus with respect to a particular target.

6.1.4 Image Synthesis

The arguments to DIRSIG are contained in a series of parameter files. These files contain the values of the factors that determine the nature of the synthesized images among other information. We keep other factors constant while varying the values that make up the multi-dimensional space as needed. The details of using the Adaptive Sampling by Histogram Equalization (ASHE) algorithm to achieve adaptive sampling are shown in Algorithm 4. In summary, the ASHE algorithm attempts to equalize the histogram of the baseline ATR performance values obtained from the synthesized images. The algorithm is implemented with a MATLAB script. DIRSIG and ancillary programs that are used for synthesizing the images are also called from MATLAB.

6.2 Experiments

We synthesize images by keeping all other factors that contribute to variation in the imaged scene constant while varying the three factors identified in Section 6.1.2. We synthesize a set of images using a random combination of these factors, and another set using combinations of factors that are evenly spaced within their possible ranges. These are compared to the set of images generated by the set of factors determined by the ASHE algorithm. The following are used in the active walker model in implementing ASHE: $N_{\text{aw}}= 5$, $LSP= 0.3$, and $SSP= 0.04N_{\text{aw}}= 5$. These are based on the results from our analysis in Sections 4.3 and 4.6. Each of the sets consists of 125 images, of spatial size 128×128 pixels, and 44 equally spaced spectral bands spanning $0.35 - 1.0$ nm. Each image took about 26 minutes to synthesize on a Linux workstation with a 3.2 GHz Pentium IV processor.

The baseline ATR performance values are also computed for the sets of images syn-

Algorithm 4 . Synthesizing hyperspectral images using the ASHE algorithm

Initial definitions:

Objective function - Baseline ATR performance
Factors that the Objective function is dependent as identified in Section 6.1.2
Range and possible values that these factors can take, also listed in Section 6.1.2

Sampling initialization:

Obtain initial random locations in n-dimensional space using active walkers
Synthesize images for combination of factors from these locations
Compute Baseline ATR performance from initial sample image points
Compute normalized histogram from initial sample performance values
Compute Overall Fitness Criterion OFC

While no. of synthesized images $\leq$ required number of images **do**
For all active walkers **do**
Obtain new sample point in multi-dimensional space
If location has already been sampled
Obtain alternate close sample point
end if
Synthesize new image based on active walker position
(*DIRSIG arguments are coordinates of active walker position*)
Add new image sample from active walker to existing images
Compute baseline ATR performance for new image addition
Compute new normalized histogram of performance values, and
Compute New Fitness Criterion NFC
If $NFC < OFC$
Single walker takes *short* step size in random direction
Else Single walker takes *long* step size in random direction
End if
End for
Compute new overall normalized histogram
Compute OFC
End while

thesized by a random combination of these factors, and those synthesized using combinations of factors that are evenly spaced within their possible ranges. These image sets are then compared to the adaptively synthesized images on the basis of representation across the range of performance values. This is determined as the range between the minimum, that is zero false alarm rate, to the maximum of all performance values recorded from the three methods used for image synthesis. By representation, we refer to each bin having at least one image so that an ATR algorithm test on the database would have considered all levels of difficulty. The images are also considered based on the distribution among the different levels of difficulty. That is, a measure of the uniformity in the distribution of images across the different levels of difficulty so that ATR algorithm tests are not biased by over-representation in a particular category of difficulty.

Figure 6.2 shows histograms indicating the spread of representation over the defined baseline ATR performance range, and the levels of representation for each performance value. There are 106 possible performance values in the range. As shown by the count of the number of bins with at least one image representation, the image set generated using the adaptive algorithm show representation of more ATR performance values than the other two methods. Note that none of the methods produce images that have performance values between 0 and 33. This is due to the threshold value used to determine the false alarm rate for the images. A higher value will result in lower baseline ATR performance values for all three methods.

Also, a comparison of the normalized versions of these histograms to a normalized uniform distribution with the same number of bins, shows that there is a more even distribution of the ATR performance values from the image set obtained using the ASHE algorithm. We use the fitness criterion given in (3.2) as an objective measure of this. Thus, the lower the value of the deviation, the more the distribution tends towards the

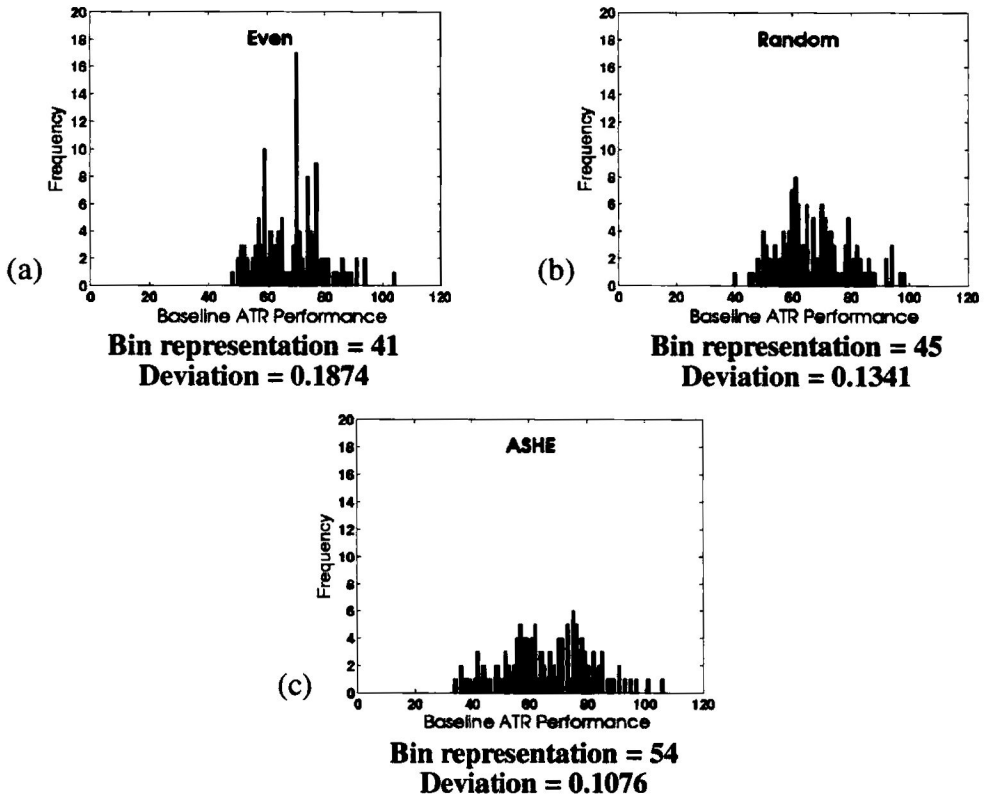


Figure 6.2: Distribution of baseline ATR performance values. Representation for images obtained from (a) combinations of evenly spaced factors, (b) random combination of factors, and (c) combination of factors obtained based on the ASHE algorithm. The bin representation is the count of bins that have at least one image, there are 106 bins in all. The deviation values are computed in a similar manner to the fitness criterion described in (3.2) earlier.

uniform. The lack of representation in the range of values between 0 and 33 diminishes the improvement recorded by using the ASHE algorithm. This is noted in the recorded deviation values. Excluding the range in the computation will make the recorded improvement more apparent.

CHAPTER 7

DEVELOPMENT OF A MEASURE OF CLUTTER FOR HYPERSPECTRAL IMAGES

In this chapter, we present our main application, which is the development of a measure of clutter for hyperspectral images. An image is said to be cluttered if some of its background and other objects may be mistaken by an Automatic Target Recognition (ATR) algorithm as the desired target. The quantity, locations and nature of these objects will determine the clutter level in the image. Motivations for characterizing and quantifying clutter in images include:

- a need to compare ATR performance on a common objective basis [83],
- the need for a measure to form the basis for a pre-processing step to discard images, or make a decision on further processing,
- the need for a measure to form the basis for a post-processing step to determine the reliability of the result of running an ATR on a scene, and
- the inverse-problem problem of creating clutter on ground scenes. e.g. camouflaging.

Such a measure of clutter will indicate the inherent difficulty for an ATR algorithm to detect targets. That is, a means to determine the degree of difficulty to detect and identify a target in a scene.

Since it is difficult to capture the multifaceted nature of image clutter in a single number, our aim is to obtain bounds on the performance of any ATR on a scene based on

a general clutter quantification scheme. That is, a high value of this quantity will indicate that any ATR will produce a high false alarm (FA) rate. A low value may, however, not result in a low FA rate. This will depend on the exact nature of the ATR.

Previous works attempting to characterize or quantify clutter in images include [94, 66, 58, 45, 51, 91, 81]. However, all these works focus on deriving clutter measures for single-band images. To the best of our knowledge, no research effort has addressed the problem of deriving a clutter measure for complete hyperspectral images.

Next, we describe our approach for developing this measure in its general form. Then, we present results from obtaining the measure for single band images, and for multi-band hyperspectral images. We also present specific applications of the derived measure in both cases.

7.1 Clutter Complexity Measure

In its general form, our approach is to obtain an aggregation of statistical image features or metrics that correlates best with baseline ATR performance. We use the terms 'features' and 'metrics' interchangeably. We compute metrics that fulfill the following criteria from the images:

1. Descriptive of scene parametric variation and significant for ATR performance.

2. Computing them only requires *a priori* information on the order of spatial extent of the target in the scene at the most.

3. Algorithmically uncomplicated, and easy to implement.

These are similar to the requirements listed in [66]. We then obtain a value indicative of baseline ATR performance from the images, and obtain the measure as an aggregation

of these metrics that correlates best with this performance. We call the derived value the Clutter Complexity Measure (CCM).

The process of combining these metrics to yield the required result is obtained through a training process on a subset of available image data. Once established, this is generalized over the complete dataset. This training process requires image data in numbers that are statistically significant. As stated earlier in Section 5.2, the availability of these images is limited. Thus, we synthesize test images as described in Chapter 5. Generalization of the derived measure from a random subset of images requires that there is a good representation of the values indicative of the ATR performance in each subset. It also requires that the range of these values is represented in the test images. Fulfillment of these requirements is improved by synthesizing images based on the ASHE algorithm. This is described in Chapter 6.

7.2 Clutter Complexity Measure for Single Hyperspectral Bands using Real Data

An ATR can utilize a combination of the information in the separate bands that make up the hyperspectral image of a particular scene. Intuitively, using multiple bands of the same scene for the purpose of target detection should yield fewer false alarms for the same probability of detection P_d when compared to using a single band. The computational resources needed by the ATR increases with each additional band, resulting in the need for an efficient selection of the bands utilized by the ATR. It will be beneficial to be able to select the bands that contain the required target information surrounded by clutter of low complexity. These fewer bands can then be used in the multiple band detection with results comparable to using all the available bands.

The clutter complexity measure (CCM) of a band can indicate the bands utility for detection. That is, bands are prioritized by their clutter complexity measure. Thus, an L

band detector will use the L bands with the least clutter complexity in the hyperspectral cube. In the following sections, we describe the process of obtaining a CCM for the bands in a hyperspectral image, and present results from experimenting with the derived CCM. The test images for the single band analysis are real, forward looking infrared (FLIR) images.

7.2.1 Baseline ATR Performance

We establish the required baseline ATR performance described in Section 7.1 by using the RX algorithm [68]. This is an anomaly detector that is capable of integrating data for multiple bands. In summary, the RX algorithm determines how much a region is different from its surrounding region relative to the arithmetic mean and variance of the pixels in this surrounding region. For each pixel, the RX algorithm computes a statistic given by:

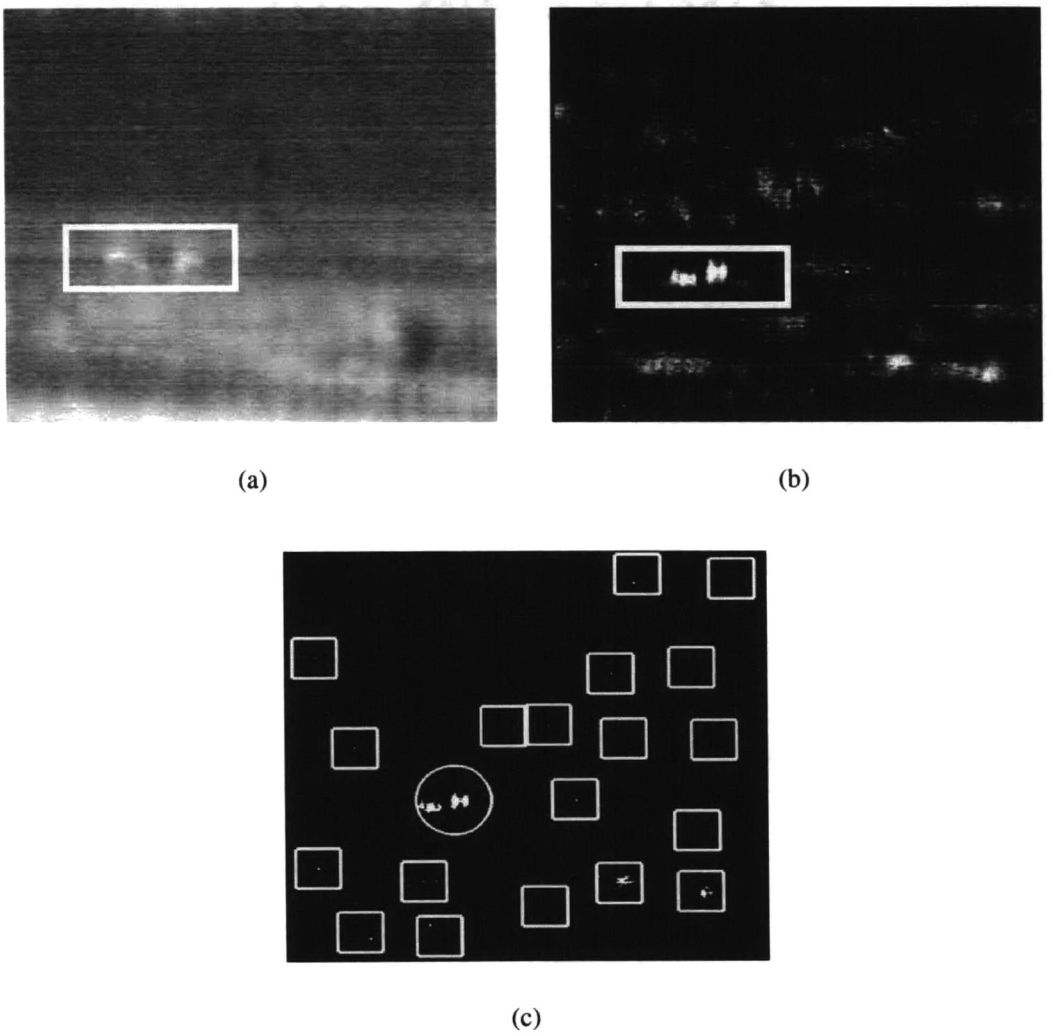
$$S = (\mathbf{x} - \mathbf{u})'\mathbf{R}^{-1}(\mathbf{x} - \mathbf{u}) \,, \tag{7.1}$$

where

$$\mathbf{u} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i, \qquad \mathbf{R} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})(\mathbf{x}_i - \mathbf{u})',$$

and $\mathbf{x}_i$ is the vector of pixels from a surrounding annular ring of length N. The computed statistic S in (7.1) is then compared to a threshold to determine the presence of a possible target.

Figure 7.1(a) shows an HMMWV military vehicle at a stand-off of 1.2 km in a forward looking infrared (FLIR) scene and Figure 7.1(b) shows the result of running the RX detector on the scene. The white portions in Figure 7.1(b) indicate high values of S and black low values. These patches are clustered together for the purpose of detection and counting false alarms. The threshold of the RX algorithm is set so that $P_d = 1$ and



(a) (b)

(c)

Figure 7.1: RX detection in FLIR images: (a) Original image (b) Image of RX statistic (c) Resulting detection image.

Table 7.1: List of some of the image statistical features used in deriving the clutter complexity measure.

Feature Name	Description
FBM Hurst Parameter	Texture roughness
Standard Deviation	Global standard deviation
Schmieder Weathersby	Average local standard deviation
Homogeneity	Average pixel variation
Energy	Average histogram energy
Entropy	Average histogram entropy
Target Interference Ratio	Average contrast
Outlier Ratio	Average percentage of outliers

the false alarm count is minimized for all the experiments. The vehicle was detected in the region marked with a circle in Figure 7.1(c). The other clustered regions that contain S values greater than or equal to the threshold are marked with squares in the detection image, these constitute false alarms.

7.2.2 Multiple-feature CCM

We obtained a clutter complexity measure as a weighted sum of statistical image features. In [45], the measure was formed by the eight features listed in Table 7.1. In addition to these, we also used five variations of the Gaussian based decomposition of images obtained by analysis-by-synthesis [8] resulting in the use of 13 statistical image features in all. We attempt to obtain the weighted sum of these 13 image features that correlates best with the performance of the RX algorithm. We computed the RX false alarm counts over a set of training images containing a given target. The false alarm count resulting from these were trained for a partition of FLIR images. Then, a set of weights were obtained that resulted in the best correlation between these false alarm counts and weighted sums of the image features. This approach is similar to that in [45] with the exception of

using the single-band RX algorithm instead of template matching to determine the ATR performance bounds.

In order to evaluate the generalization of the derived weights, we would need to obtain comparable results from them on a different partition. The work in [45] showed that such weights are not independent of the target in an image so the target in the partitions have to be the same. Given the same target, if other objects in the scene are altered, the clutter complexity should still be able to predict the ATR performance. As a result, the clutter complexity measure should yield good correlation with ATR performance bounds for a disparate set of test images that include the same target object. Our experiments only obtained such weights resulting in a good correlation on a per partition basis. That is, our training and test data set were the same. This is due to the limited number of real image data that was available for the training process described earlier. Hence, the use of synthesized hyperspectral images, described in the subsequent experiments.

7.2.3 Single-feature CCM

To avoid the questions raised by the inadequate training to obtain required weights, we tested each of the statistical image features that made up the weighted sums to see if any of them had good correlation to the false alarm count rate for all the images. Such correlation in all the image sets suggests that the statistical image feature is a good indicator of complexity. The important distinction between this and the multiple-feature clutter complexity measure is that there is no need for training. We chose the feature with the best average correlation to the false alarm count rates for all the images.

7.2.4 Single-Band CCM Experiments

Our test data set comprises of 5, 28-band (*cubes*) of forward looking hyperspectral images taken with the same polarization of 90^0 and wavelength ranging from 460 – 1,000 nm in steps of 20 nm. Each of these 28 bands represents the same target at the same pose and stand-off distance. These criteria also formed the basis of the partitioning in [45]. Due to a computer memory constraint in running the multiple band RX algorithm, alternate bands are chosen resulting in 14 bands for the experiments. The choice of alternate bands ensures the availability of a good spread of information across the hyperspectral bands.

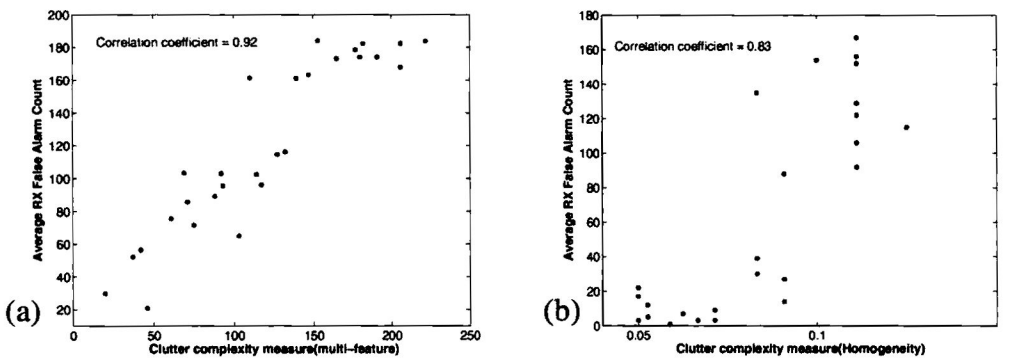
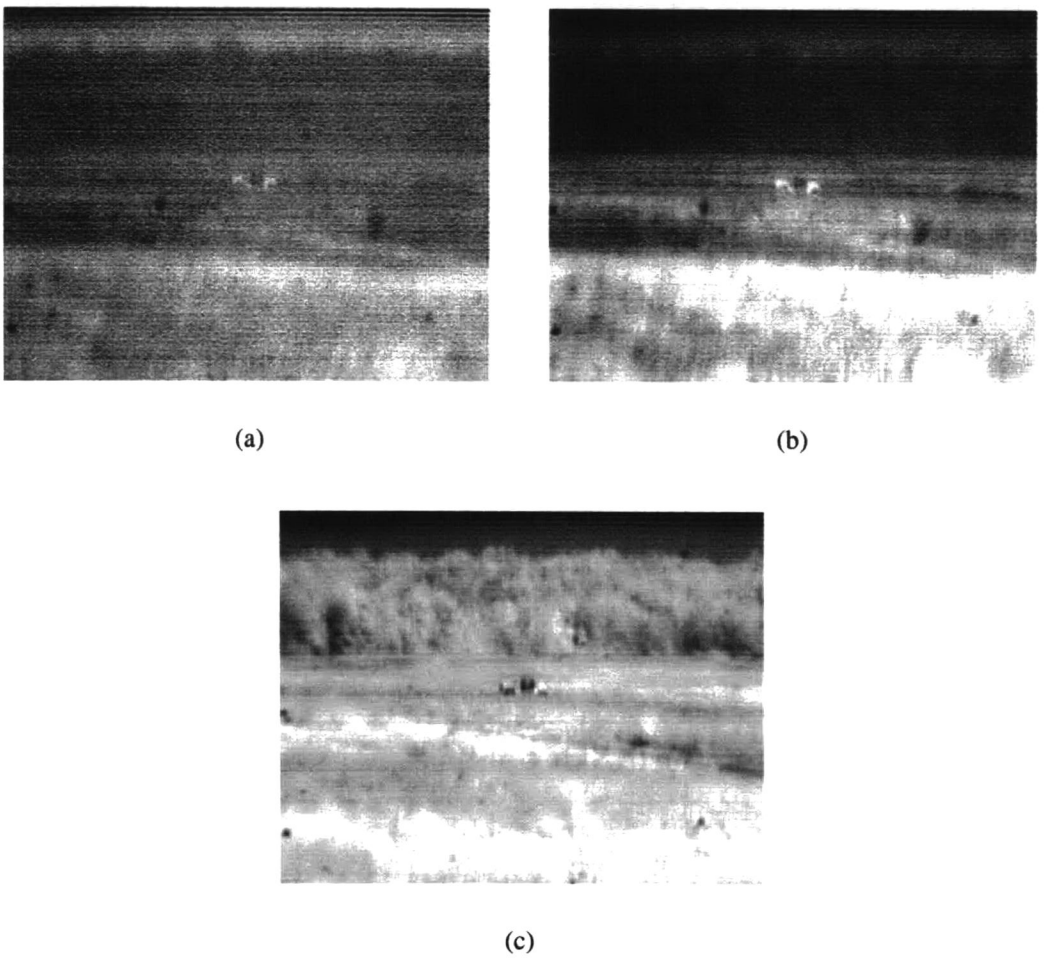


Figure 7.2: Scatter plot of clutter complexity measure and false alarm count: (a) Weighted sum of multiple features (b) Single feature.

Figure 7.2(a) shows the good correlation obtained between a weighted sum of features, i.e. the clutter complexity measure, and the false alarm count for a partition of images. Figure 7.2(b) also shows good correlation between the false alarm count and the chosen single image statistical feature (homogeneity). These results are typical for all the test data and suggest that these measures are good indicators of complexity in our test images. Figure 7.3 shows examples of images with low, medium and high complexity as classified using the weighted sum clutter complexity measure.



(a) (b)

(c)

Figure 7.3: FLIR band classification by clutter complexity measure: (a) Low clutter complexity number = 38.84 (b) Medium clutter complexity number = 73.56 (c) High clutter complexity number = 112.91

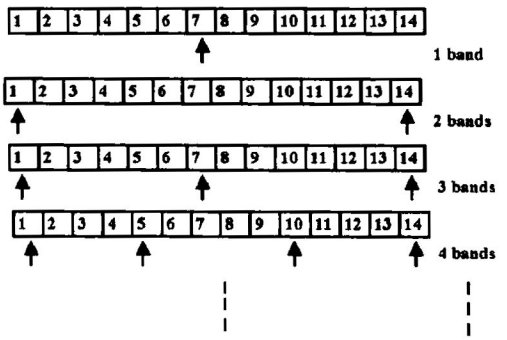


Figure 7.4: Example of band selection based on wavelength to ensure a uniform distribution of the choice of hyperspectral bands.

To test the utility of the clutter complexity measure for band selection, we start running the RX detector using a single band. More bands are then added with the choice of each extra band based on one of the following: clutter complexity measure, derived from (1) single, (2) multiple features, and (3) wavelength. The ordering by wavelength is done in order to ensure a uniform spread of the choice of bands over all available wavelengths as shown in Figure 7.4.

Figure 7.5 shows a plot of the average false alarm count over the five *cubes* of hyperspectral data against the number of bands used by the ATR for three scenarios over all 14 bands. To obtain an optimal subset of k bands, all the possible combinations of the 14 bands are considered. This was done for $k = 1$ to 5 resulting in $3,472$ ATR experiments for each set. A plot of the average false alarm count for the optimal choice of 1 to 5 bands is also shown in Figure 7.5. The probability of detection (P_d) was set to 1 for all experiments. The false alarm counts shown are obtained by averaging over the five hyperspectral images used for the experiment.

As expected, the false alarm count reduces as more bands are added for all experiments. This shows that the information in multiple bands of the hyperspectral data are complementary. The target information adds up more rapidly than the information in the

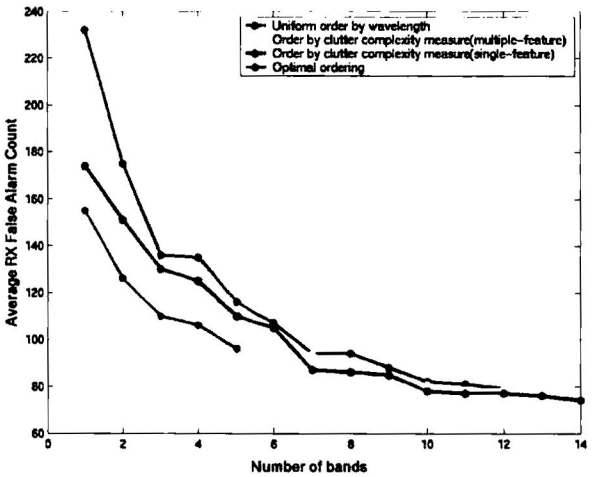


Figure 7.5: Performance of clutter complexity measures represented as average false alarm count versus number of bands integrated into the RX detector.

surrounding clutter, resulting in fewer false alarm counts for the same P_d.

The average false alarm count is less for the bands ordered using our derived clutter complexity measure compared to when the bands are ordered by wavelength. This indicates that the clutter complexity measure criteria for band selection results in an improvement in the performance of the ATR. How much of an improvement is seen by comparing the result of ordering on the basis of the clutter complexity measures to the optimal ordering. The false alarm count for the optimal ordering is about 30% less than ordering by the clutter complexity measures for 1 to 5 bands. This is about the same improvement of the ordering by clutter complexity over the uniform ordering by wavelength which does not take any clutter information into account.

There is a rapid drop in false alarm count for all the experiments from 1 to 3 bands. A *knee* is seen when between 4 to 7 bands are utilized by the ATR and there is little improvement after the use of 7 bands. The basis for ordering the bands becomes less important as more bands are added for the detection process beyond 8 bands. The curves

in Figure 7.5 merge as expected when all 14 bands are used.

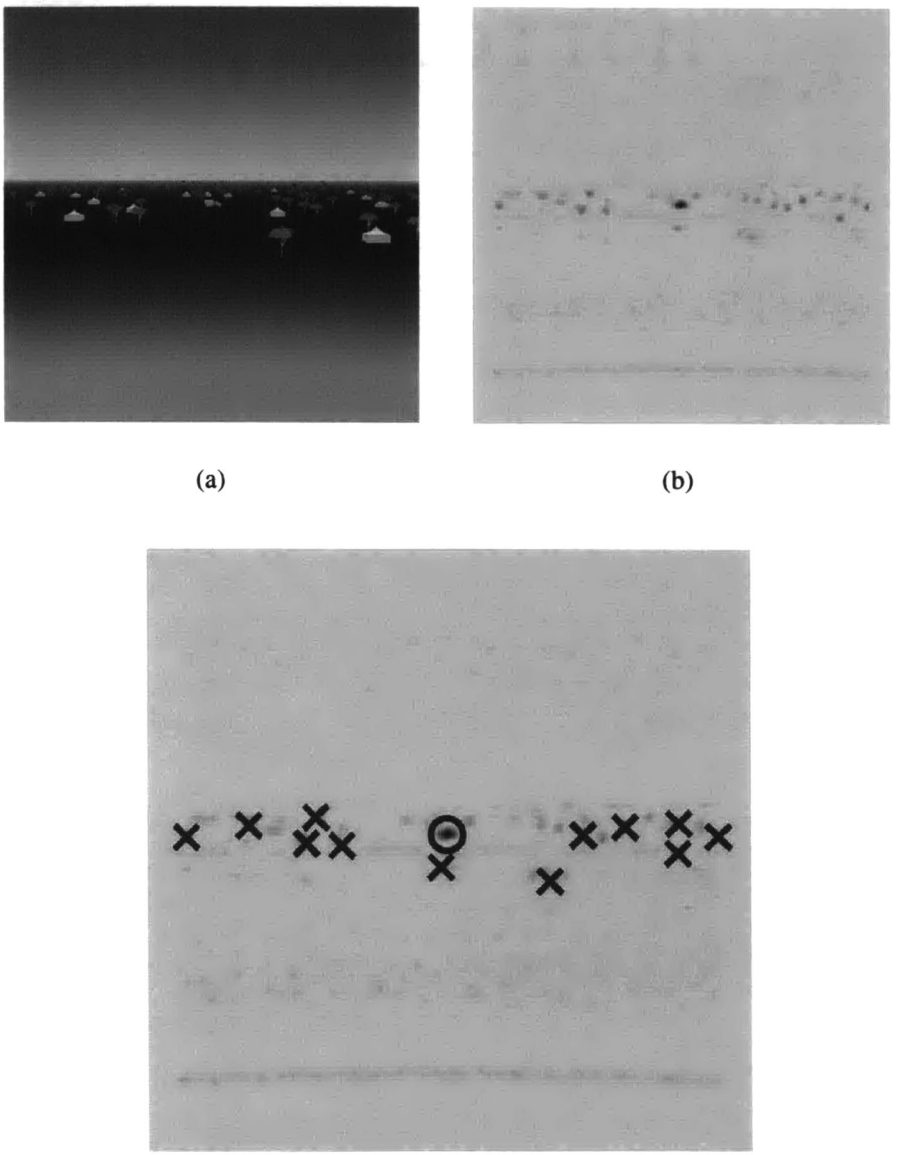
The derived CCM for single bands is shown to be a useful criterion for choosing bands in a multi-band ATR detection. It is noted that the performance of the single-feature and multi-feature clutter complexity measures are comparable. Clutter complexity measures derived from relevant multiple features should generally be more reliable than that from a single feature because of the usual multi-faceted nature of clutter. The next set of experiments reports work on deriving such multiple-feature clutter complexity measures for complete hyperspectral image *cubes*.

7.3 Clutter Complexity Measure for Hyperspectral Images

The previous experiments established the feasibility of our approach to obtain a CCM. Our goal is to establish such a measure for complete hyperspectral images. We follow the same approach outlined in Section 7.1. Here, we use synthesized hyperspectral infrared (HSI) images as our test data. We are able to follow the described training process because we have synthesized images in statistically significant numbers. We describe the process for deriving a CCM for hyperspectral images, and present subsequent experiments and results.

7.3.1 Baseline ATR Performance

We establish ATR baseline performance by utilizing an idealized implementation of a normalized, multispectral matched filter ATR via the Adaptive Coherence Estimator (ACE). This was described fully in Section 6.1.3. Its application in deriving a baseline ATR performance is similar. Figure 7.6 shows one of the bands from an example synthetic hyperspectral image *cube*, the resulting ACE statistic image, and the final detection image after thresholding. The statistic image is on a gray scale, with black (0) - detection



(a) (b)

(c)

Figure 7.6: Target detection in HSI using ACE: (a) Band ($\lambda = 8.40$ microns) from HSI image (b) Image of ACE statistic (c) Detection image, with 'O' representing the target and 'X' the false alarms.

with certainty, and white (1) - no detection at the two extremes. The target was detected in the location marked by the 'O' and the 'X's indicate false alarms in Figure 7.6(c).

7.3.2 Image Clutter Metrics

The utilized image features or metrics also fulfill the requirements outlined earlier in Section 7.1. They fall into two broad groups of features derived from hyperspectral single bands, and those derived from the complete hyperspectral cube.

Metrics Derived from Single Bands

The image clutter metrics that were used in [62] and [23] were mostly based on statistical features of the images. We implemented these and computed them for each band of the hyperspectral images. In addition to these, we also computed a metric based on parameters derived from Gabor filtering of the hyperspectral image bands. The Gabor filter extracts edges from an image at different orientations [85]. Two parameters are derived from these filtered images: the first parameter, p, is an indication of the distinctness and frequency of edges in the filtered image, while the second, c, is related to the range of pixel values in the image. All these fall under the category of single-band clutter metrics. To extend these for hyperspectral images, we compute distribution representative values like maximum, minimum, mean, median, and range for each metric resulting in five hyperspectral clutter metrics derived from each single-band metric.

Metrics Derived from Hyperspectral Image

Image clutter metrics were also computed directly from the hyperspectral image *cube*. A metric was derived from the correlation between the hyperspectral bands in an image. Generally, lower correlation between the bands signifies more unique information in each

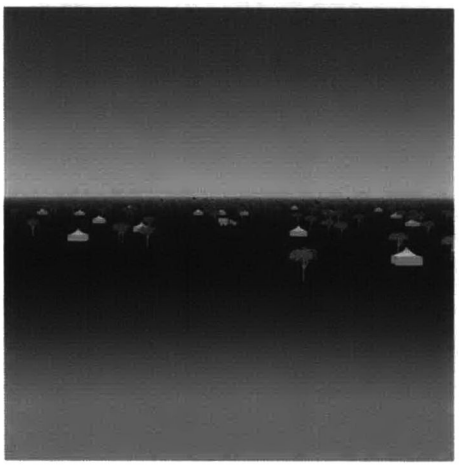
band, resulting in the better performance of multispectral ATR algorithm.

Two metrics were also computed using the vector of the pixels in the spatial dimensions along the spectral dimension. The length of the vector is equal to number of bands. One of the metrics we computed was based on the dot product between a pixel vector and the surrounding pixels. A high value indicated that the pixel vectors are from a homogeneous region, and dissimilar otherwise. The other one was based on the Kullback-Leibler distances which is the relative entropy between a pixel and its surrounding pixels [82]. Hereby, each pixel vector is modeled as a distribution, and the distance is a measure of the difference between a pixel and another. Thus, pixels in homogeneous regions will result in lower values for this metric. Finally, we derive a set of image clutter metrics from Gray Level Co-occurrence Matrices (GLCM) as proposed in [35]. This method has been used for texture characterization in images [36]. We extend the spatial-spatial offsets implemented in single-band image processing into the spectral dimension. We also experiment with a variant of the GLCM in which, the pixel locations are randomly chosen over the whole hyperspectral image cube. The five metrics derived from each variant of the GLCM are maximum value, energy, entropy, contrast and homogeneity. A more detailed description of the clutter metrics is contained in Appendix A.

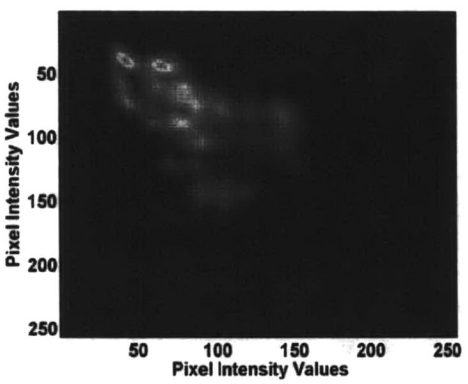
A summary of the image clutter metric categories and brief descriptions are shown in Table 7.2. We implemented a total of 129 clutter metrics. We show some examples of these metrics in Figures 7.7 and 7.8. The hyperspectral image clutter metrics derived from these were described earlier.

7.3.3 Determining Significant Metrics

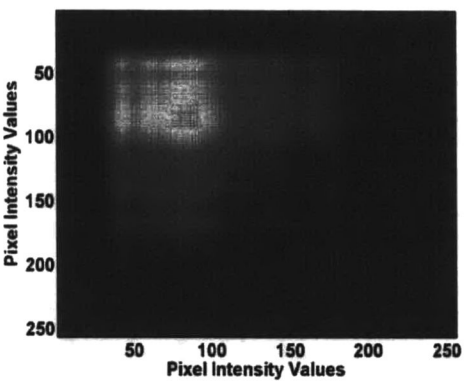
A factor analysis scheme is implemented to remove clutter metrics that are not significant for ATR performance, and to reduce redundancy among the remaining. This will result in



(a)



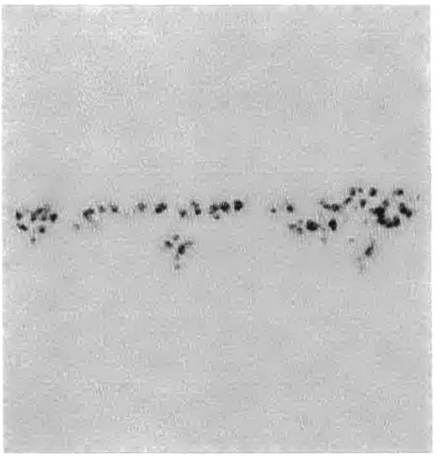
(b)



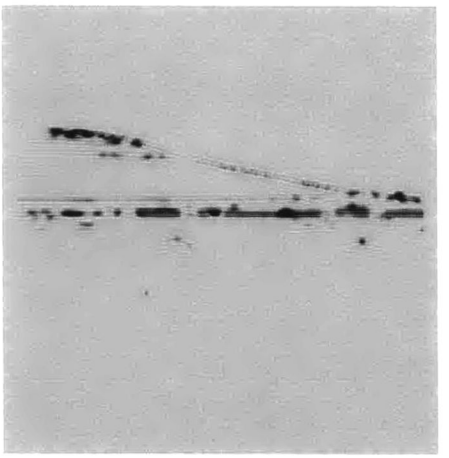
(c)

Figure 7.7: Gray Level Co-occurrence Matrices from hyperspectral images: (a) Single band from test image, (b) GLCM using offset based on target size and (c) GLCM using random pixel locations.

(a)



(b)



(c)

Figure 7.8: Gabor filtered band from hyperspectral image: (a) Hyperspectral image band, (b) Gabor filtered image, filter at $15°$ orientation, extracts near vertical edges, and (c) Gabor filtered image, filter at $90°$ orientation, extracts near horizontal edges.

Table 7.2: Summarized list of clutter metrics used in deriving the clutter complexity measure for hyperspectral images.

Metric Name	Description	No. of Metrics
Single-band clutter metrics[1]		
Standard deviation	Global standard deviation	5
Schmieder Weathersby	Average local standard deviation	5
Homogeneity	Average pixel variation	5
Energy	Average histogram energy	5
Entropy	Average histogram entropy	5
Target Interference Ratio	Average contrast	5
Outlier Ratio	Average percentage of outliers/edges	5
FBM Hurst Parameter	Texture roughness	5
GGABS(5 variations, $I - V$)	Generalized Gaussian Analysis-By-Synthesis	25
Gabor filter(5 orientations)	Parameters p (edge content), c (pixel intensity range)	2×5×5=50
Derived from band information content		
Band correlation	Mean/Median correlation in HSI bands	2
Anomaly detectors		
DotProduct	Average dot product of pixel vectors	1
Kullback-Leibler	Average relative entropy of pixel vectors	1
Derived from GLCM[2]		
GLCM Imax.	Inverse of maximum value from matrix	2×1=2
GLCM Energy	Energy computed from matrix	2×1=2
GLCM Entropy	Entropy computed from matrix	2×1=2
GLCM Contrast	Contrast computed from matrix	2×1=2
GLCM Homogeneity	Homogeneity computed from matrix	2×1=2
	Total	129

[1] 5 metrics - Min., Max., Mean, Median and Range are computed from the distribution obtained from computing these from the HSI image single bands
[2] Same values computed for both implemented variants of GLCM described

a reduction in the dimensions of the clutter metrics space, and a reduction in the required number of operations to compute them. The aim is to reduce the dimensionality yet retain significant information about clutter in the images in the clutter metrics space. In contrast to Principal Component Analysis (PCA) [50], in which the resulting dimensions in a reduction process do not map directly into the original space, our factor analysis algorithm allows the identification of the retained dimensions from the original space. This is shown in Algorithm 5.

Algorithm 5 . Factor Analysis to Determine Significant Metrics

Randomly select images of required number from database
to form a training set

for all $\alpha \in$ set of clutter metrics **do**
    compute $|CC(\alpha,$ false alarm rate (FA)$)|$
    discard α from the set if CC is
    'insignificant' i.e. ≤ 0.5
end for

compute correlation matrix of the remaining metrics

for all combinations of α and β of the remaining metrics, **do**
    if $|CC(\alpha, \beta)|$ is 'significant' i.e. ≥ 0.8 **then**
      if $|CC(\alpha, FA)| > |CC(\beta, FA)|$
        discard β
      else
        discard α
      end if
    end if
end for

where CC(x,y) is the correlation coefficient between variables x, and y

7.3.4 Hyperspectral Images CCM Experiments

The clutter metrics computed for each hyperspectral image are normalized for all images to avoid a bias in further processing results due to large ranges of absolute values from metric to metric. We employ linear regression to obtain a weighted combination of the subset of image clutter metrics that correlate best with the baseline clutter levels represented by false alarm rates. This weighted sum is the clutter complexity measure (CCM). A high correlation coefficient (CC) will indicate that the CCM is a good indicator of the baseline clutter levels, that is, monotonically related to ATR task difficulty. The correlation coefficient is the normalized measure of covariance between false alarm rate and the computed clutter complexity measure, and serves as our performance measure.

Data Description

We experimented with two sets of images. The first set consisted of 216 synthesized hyperspectral infrared images.he process of image synthesis, and the image specifications are described in Chapter 5. The target template under test are of size $9 \times 9 \times 126$ pixels. These are averaged as described in Section 6.1.3 to obtain column vectors of length 126 used as arguments by the ACE filter ATR. Each image had either a truck or tank as target, and contained varied clutter at varying levels. Chapter 5 also shows example images, and targets of interest.

The second set consisted of 125 images, synthesized based on the ASHE algorithm as described in Section 6.1. Each image in the set is of spatial size 128×128 pixels, and 44 equally spaced spectral bands spanning $0.35 - 1.0$ nm. The target template under test is of size $3 \times 3 \times 44$ pixels, and the location is indicated by the arrow in the scene template shown in Figure 6.1.

Procedures and Results

We conduct similar experiments with both image sets. Our first sets of experiments sought to obtain the subset of metrics that result from the described factor analysis algorithm. To achieve this, we made random selections of images, and record the resulting clutter metrics subset. This subset results from the factor analysis of all the computed clutter metrics from the images. This process is repeated $1,000$ times, each using a unique combination of images of the same number. By applying linear regression, a weighted combination of these clutter metrics which correlates best with the false alarm rates in the selected images is obtained. The selected images thus serve as a training sample set and the obtained weight from the training process is applied to the remaining images, which serve as the test set. We experimented with different training image sample sizes - from 5% to 40% of the total database size, that is, sets of 11 to 86 images. In each case, all the unselected images serve as the test set.

Figures 7.9(a-h) show histograms of the occurrences of the clutter metrics in the selection process using the first image set. The smaller sets do not show a clear dominance in terms of occurrence of any particular metric. As the training sample set size increases, for example, at 20% of the database size, there is a clear increase in the frequency of a few of the metrics while many others do not occur at all. This trend continues as the training sample set size is increased.

Table 7.3(a) shows the average values of the correlation coefficient for different training sample set sizes. Results from training with smaller set sizes show a perfect correlation between the computed CCM with false alarm rate for the test image sets, but relatively poor generalization to the whole database. This signifies an over-training. Increasing the train dataset alleviates the over-training problem and improves on the gen-

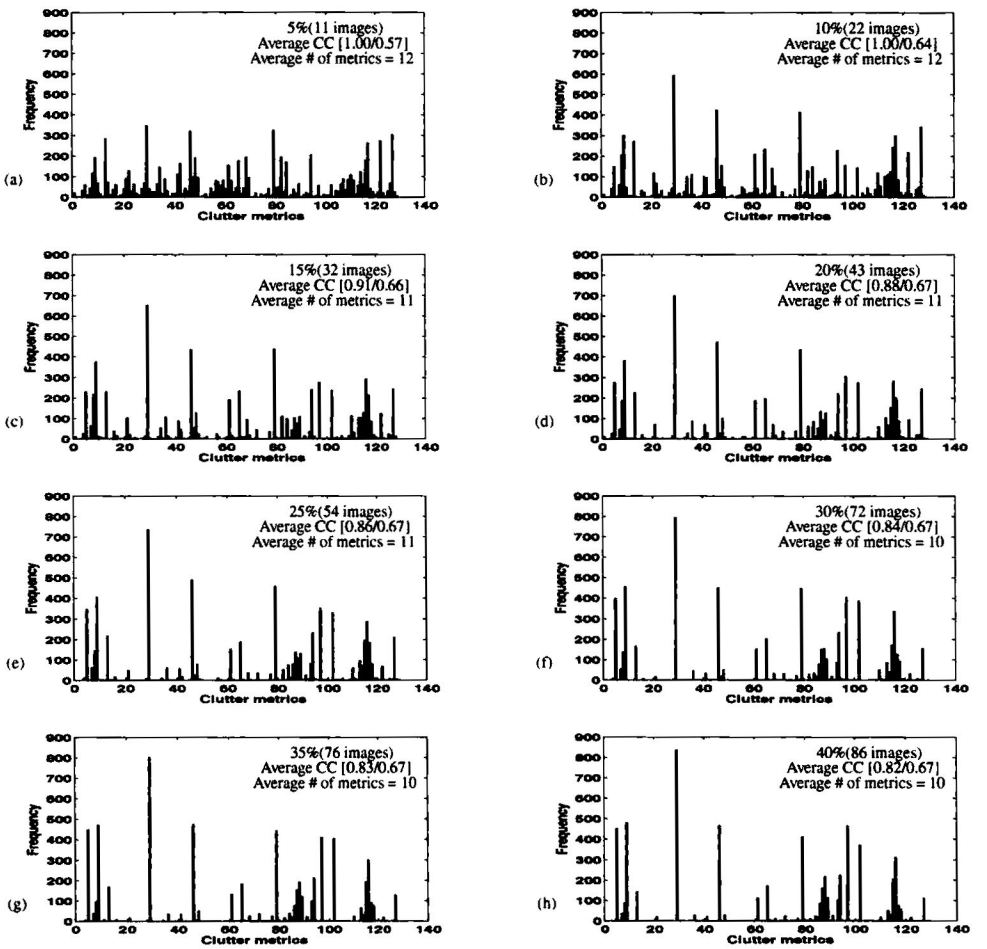


Figure 7.9: Frequency of selection of clutter metrics for training image sets ranging in sizes from 5% to 40% of entire database (11 to 86 images). Noted on the plots are the average number of selected metrics (rounded to the nearest integer), and the average CC values e.g. [0.82/0.67], which are the average CC for the training set and the test sample set respectively. The numbers in the abscissa represent an arbitrary but consistent indexing of the clutter metrics.

Table 7.3: Averaged correlation coefficients obtained between the clutter measure and false alarm rates using different training sample sizes.
(a) Using the subset of clutter metrics selected by the factor analysis process

Training partitions	Test partitions	
Sample size	Same as train sample	Test sample
11 images (5%)	1.00	0.57
22 images (10%)	1.00	0.64
32 images (15%)	0.91	0.66
43 images (20%)	0.88	0.67
54 images (25%)	0.86	0.67
65 images (30%)	0.84	0.67
76 images (35%)	0.83	0.67
86 images (40%)	0.82	0.67

(b) Using a further subset of the metrics used to generate the results in Table 7.3(a) - only the eight metrics with the highest overall frequencies

Training partitions	Test partitions	
Sample size	Same as train sample	Test samples
11 images 5%	0.84	0.40
22 images (10%)	0.77	0.57
32 images (15%)	0.76	0.62
43 images (20%)	0.74	0.64
54 images (25%)	0.73	0.65
65 images (30%)	0.72	0.66
76 images (35%)	0.73	0.66
86 images (40%)	0.72	0.67

Total of 1,000 experiments with first image set. Sizes are listed as percentages of the total database.

eralization. This improvement saturates with the use of about 20% of the entire database as training samples, which is 43 images in this case.

For training set samples $\geq$ 20%, only 8 clutter metrics were consistently chosen at least 30% of the times during the selection process. The ratio of the frequency of selection of these clutter metrics to all others is also generally large. The indices (arbitrarily assigned) and brief description of these 8 metrics are: #5 - homogeneity derived from the GLCM with known offset, #9 - contrast derived from the GLCM with random offset, #29 - range of the p values from the Gabor filtered images at 90° orientation, #46 - median of the c values from the Gabor filtered images at 60° orientation, #79 - range of the FBM Hurst parameter obtained from images' single bands, #97 - minimum of the homogeneity obtained from the images' single bands, #102 - minimum of the outlier/edge parameters obtained from the images' single bands, and #116 - median of the third parameter of the Gaussian decomposition of the images' single bands.

We performed further experiments with these metrics and show the results in Table 7.3(b). It shows the result of using only the combination of these dominant image metrics for obtaining the CCM for different train sample set sizes. The same trends noted and discussed in the previous experiment, in which the complete subset of clutter metrics resulting from the factor analysis algorithm are employed, is also noticed here. The correlation coefficient values are lower in some cases, this is due to a further reduction in the clutter metric space used to determine the complexity measure.

Empirical timing tests show that it takes about 8.4 minutes to compute these 8 clutter metrics from an image, compared to 75.4 minutes taken for running the ATR for the same image. Both the ATR and clutter measures were implemented in Matlab 6.0 and the tests were carried out on a workstation with a 3.2 GHz Pentium IV processor.

Correlation coefficient results obtained using the second image set are shown in Ta-

Table 7.4: Averaged correlation coefficients obtained between the clutter measure and false alarm rates using different training sample sizes.
(a) Using the subset of clutter metrics selected by the factor analysis process

Training partitions	Test partitions	
Sample size	Same as train sample	Test sample
7 images (5%)	1.00	0.29
13 images (10%)	0.94	0.42
19 images (15%)	0.89	0.62
25 images (20%)	0.88	0.67
32 images (25%)	0.86	0.69
38 images (30%)	0.85	0.72
44 images (35%)	0.84	0.72
50 images (40%)	0.84	0.74

(b) Using a further subset of the metrics used to generate the results in Table 7.4(a) - only the eight metrics with the highest overall frequencies

Training partitions	Test partitions	
Sample size	Same as train sample	Test samples
7 images (5%)	0.93	0.16
13 images (10%)	0.91	0.31
19 images (15%)	0.87	0.48
25 images (20%)	0.86	0.59
32 images (25%)	0.85	0.62
38 images (30%)	0.84	0.67
44 images (35%)	0.83	0.69
50 images (40%)	0.83	0.71

Total of 1,000 experiments with second image set. Sizes are listed as percentages of the total database.

bles 7.4(a) and 7.4(b). Note the similar trends to the previous experiments in terms of generalization of the derived CCM. Eight dominant clutter metrics were also recorded when the performance, indicated by the average CC values saturates. These are median of p values from Gabor filtered images at 120° orientation, minimum of FBM Hurst values from images' single bands, minimum, median and range of the target interference ratio from the images' single bands, maximum of the first parameter of the Gaussian decomposition of the images' single bands, and range of the second parameter of the Gaussian decomposition of the images' single bands. These metrics are different from those obtained from the initial experiments, indicating that the derived CCM is image set specific. Also, the generalization performance saturates with the use of 30% of the entire image set for training in the second experiment, compared to 20% in the first. Both fractions of the image sets result in approximately 38 images. Using this training image size, the derived clutter measure is dominated by eight clutter metrics in both cases. This indicates the the required number of training images is function of the number of dominant clutter metrics used in the CCM derivation, and not the total number of images in the experimental set.

We also show the distribution of the CC values resulting in the averages shown in Tables 7.4(a). Note that when a random selection of 38 or more training images are used, 90% or more of the CC values are ≥ 0.6. This is important because it shows that the CCM for an image set can be obtained using any random selection of training images from the complete set.

In summary, our results show a more frequent selection of a further subset of metrics used to determine our clutter measure. We refer to these as the dominant metrics. These metrics are unique for each experiment, indicating that the derived CCM is image set specific. A random set of about 38 images is shown to be sufficient to define the CCM

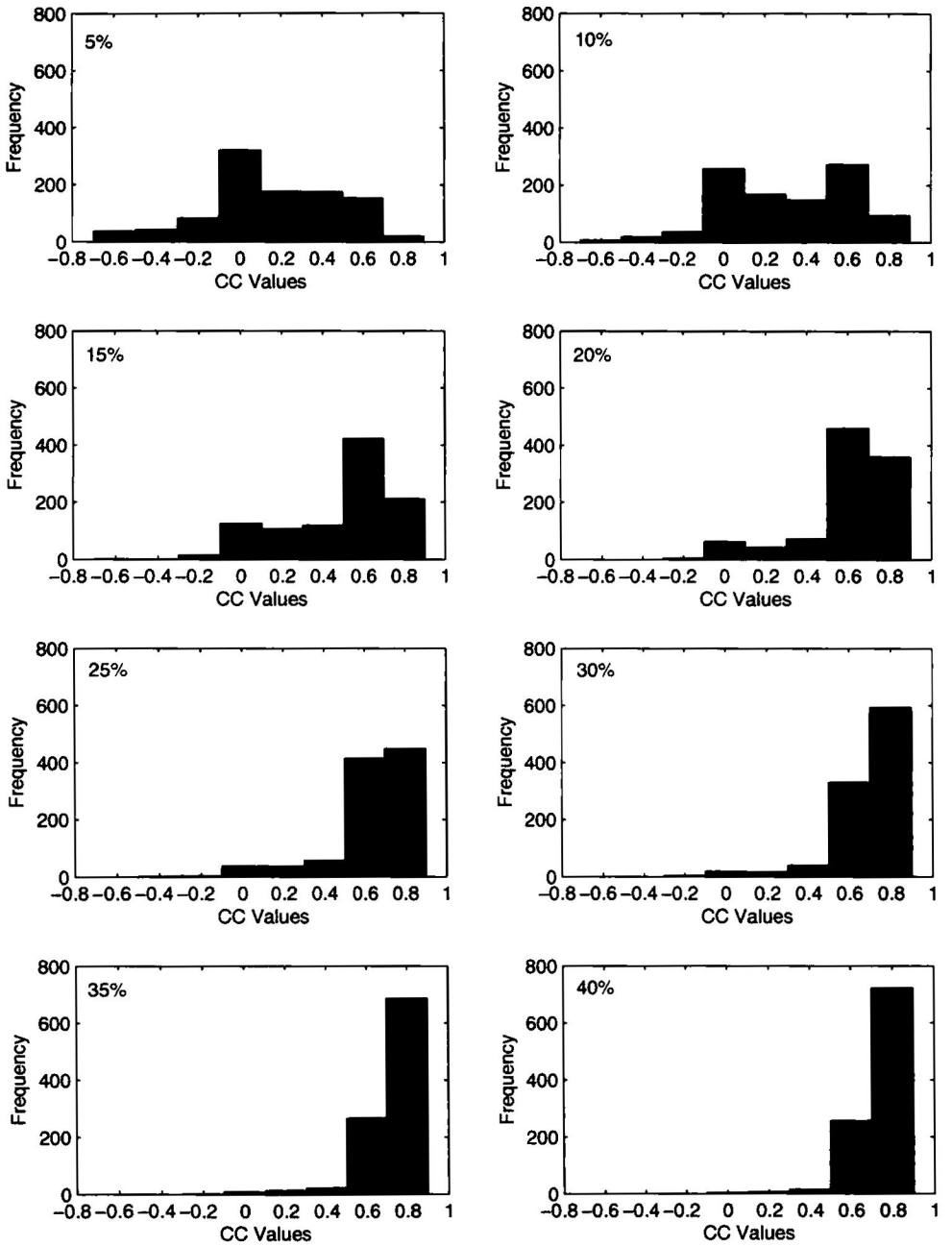


Figure 7.10: Distribution of CC values resulting in the average values shown in Table7.4(a). The indicated percentages, and the actual number of images that they represent are also shown in the same table.

using eight dominant metrics. The derived clutter measure from these training images, generalizes well for the entire database by predicting the amount of clutter in them. Further experiments to determine the clutter measure using only the dominant clutter metrics yielded similar results. Comparison of the time taken to compute the CCM from these dominant clutter metrics from and running the ATR on an image shows a ratio of about $1 : 9$ in the first set of experiments.

CHAPTER 8

CONCLUSIONS

We presented a novel, progressive adaptive sampling algorithm, called Adaptive Sampling by Histogram Equalization (ASHE). The algorithm adapts the local sampling density on a function, based on the distribution of already obtained samples. The aim of adaptive sampling is the efficient distribution of discrete samples used in representing a continuum. Efficient sample distribution reduces the inherent error that results from a sampling process. For nonstationary functions, adaptive schemes produce higher sampling densities in regions of higher complexities, that is, where the rate of change in the sampled function is higher. In numerous scientific applications, there is no prior knowledge of the local complexities in the sampled function, and the cost of obtaining each sample is prohibitive. Examples of such costs, which limit the number of samples that can be obtained, are time and computational resources. Thus, extra constraints are placed on adaptive sampling schemes. For efficient adaptive sampling, existing algorithms either require prior knowledge of the local complexities in the function, a high computational overhead, such as an acceptance or rejection step, or they require a large number of samples to converge. The ASHE algorithm requires no prior knowledge of the local variations in the sampled function. Also, it only adds a minimal overhead of computing a histogram of sample values at each step of the sampling process.

In the following sections, we summarize the main contributions of this dissertation, and discuss the findings from an application in which our developed algorithm was utilized. Finally, we make suggestions on further work.

8.1 Summary of contributions

In Chapter 2, we presented the basis of the ASHE algorithm as progressive sampling based on the distribution of already obtained samples. Typical sampling algorithms focus on the domain of the independent variables. Our focus is on the co-domain of the sampled quantity. We showed that, for a nonstationary function, even spaced sampling on the co-domain results in a sampling density that is proportional to the rates of change in the sampled function. We thus sampled in order to equalize the distribution of sample values. This results in sample densities in the domain that are proportional to the rate of change in the function, hence the adaptive sampling. To the best of our knowledge, this is a novel approach to adaptive sampling. Since the sampling scheme attempts to equalize the distribution of samples, we called it Adaptive Sampling by Histogram Equalization - ASHE algorithm. We illustrated the improved performance by the ASHE algorithm by comparing it to even spaced and random sampling. Even spaced or random sampling are the obvious options for obtaining expensive samples when there is no prior knowledge on the local complexity in a function. We identified the reasons precluding a rigorous mathematical proof of the improvement recorded by the ASHE algorithm. The most basic of these being the assumption that there is no prior knowledge on the nature of the sampled function. We however studied the algorithm further, by conducting performance and sensitivity analysis in a manner similar to those in other heuristic algorithm studies. Finally, we discussed broad areas of possible applications of the ASHE algorithm.

We introduced three stochastic optimization models in Chapter 3. These are: (1) an active walker model, based on elements of the random walk and Brownian motion, (2) an ant model, based on the simulation of foraging habits of insects, and (3) an evolutionary algorithm model, based on the simulation of natural dynamics in a population of organ-

isms. The basic forms of the models are described. We then developed three algorithms by adapting each model to implement the ASHE algorithm.

We conducted a performance and sensitivity analysis of the three models in Chapter 4. First, we established two objective measures for comparison. These are based on the frequency content, and the entropy measure of information in a sampled function. Both measures are designed to have a positive correlation with increase in function complexity. Our measure of performance is defined as the correlation coefficient between these measures, and the sample density obtained from each model. A high positive value (maximum $= 1$), indicates a good performance. We identified factors that could affect the performance of each model, and recorded their performances for varying values of these factors. Comparing the best performance of the three, the ant and evolutionary algorithm models performed marginally better than the active walker model. More importantly, the active walker model showed a correlation between the individual factors, and the performance. This is a crucial requirement for heuristic algorithms. If this is not met, the algorithms are ad hoc, requiring customization for each application. The other two models contained one or more factors that showed no individual correlation with the sampling performance. This limits their practical use. Based on our findings, we studied the active walker model further, by considering its scaling properties. Our results indicated that the model performance does not change appreciably with change in the dimensions of the sampled space.

We utilized the ASHE algorithm in the synthesis of hyperspectral images. The availability of real images of these types is limited, and synthesized images are used in their place. For our purpose, we required images that are diverse with respect to Automatic Target Recognition ATR performance. In general, image synthesis is computationally expensive. Also, there is no prior knowledge of how the factors in the image synthesis

process affect the ATR performance. We described the nature, uses, and synthesis of hyperspectral images in Chapter 5. We then synthesized images using a combination of even spaced factors. In Chapter 6, we employed the ASHE algorithm in the image synthesis process, and compared the images to those synthesized using even spaced, and random placed factors. This comparison is based on the requirement of image diversity with respect to ATR performance. Our results showed a marked improvement over the other methods. The worst performance was recorded for images synthesized using a combination of even spaced factors.

In Chapter 7, we developed a framework for quantifying clutter in hyperspectral images. By clutter, we mean any object or structure in an image that inhibits the detection of a target of interest. We derived this measure as an aggregation of image features that correlates best with ATR performance bounds. We called this the Clutter Complexity Measure CCM. This is an indication of the inherent difficulty for an ATR to identify a target in a scene. It is however, not based on any particular ATR, thus making it a good objective basis for comparing the performance of disparate ATRs. Our initial experiment to investigate the feasibility of this approach used single bands from real hyperspectral images. Our results showed that CCM derived for this images was useful in the efficient ordering of hyperspectral bands in a multi-band detection scheme. Using the band selection based on the CCM for the multi-band detection, we recorded an average of 30% improvement over the even spread band selection. We also successfully derived a clutter complexity measure for complete, synthesized hyperspectral images. In computing this, we developed 129 image features, and computed the CCM as an aggregation of a subset of these features. We obtain the subset of features through a factor analysis process. We were able to derive a CCM using any random selection of images from the complete set. We determined that the required size of the selection is dependent on the number of im-

age features aggregated to compute the CCM. In our experiments, the CCM consisted of 8 image features, and this required about 40 images. We also determined that the derived CCM is specific to an image set. The CCM derived for complete hyperspectral images was computed in 11% of the time it took to compute a baseline ATR performance. The CCM was shown to accurately predict the baseline ATR performance bounds in at least 64% of the cases.

8.2 Suggestions for further work

In the three models employed in implementing ASHE, the input from the fitness criterion is modeled as a step function in their outputs. In the active walker model for example, the fitness criterion input will result in either a long step or a short step, with nothing in between. Further work needs to be done to investigate the effect of using a different output model. That is, one in which the modeled output is a function of the amount of change in the input. A linear, exponential, or other non-linear models are examples that could be explored.

In the ASHE based image synthesis process, further work needs to be done to identify the effect that individual, and combination of factors have in the synthesized images. Factors that result in rapid image variation with respect to ATR performance can be identified using pattern analysis methods. Also, the use of a multi-dimensional objective function in the ASHE based image synthesis process needs to be investigated. This is in contrast to our use of only the baseline ATR performance. Other computationally less expensive indicators of image variability may be used to form a multi-dimensional histogram to be equalized.

Most of our test images for the experiments with the clutter complexity measure have been synthesized. The scheme to derive this measure needs to be tested using real

hyperspectral images.

APPENDIX A

CLUTTER COMPLEXITY METRICS

A.1 Single-band clutter metrics

A.1.1 Global standard deviation

$$\sigma_{\text{metric}} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(I_i - \bar{I})^2} \tag{A.1}$$

where I are the intensity values in a hyperspectral band with mean $\bar{I}$, and T is the total number of pixels in this band.

The metrics described in appendices A.1.2 to A.1.7 are computed locally. That is, each hyperspectral image band is divided into N windows, with each window containing W pixels. The size of the window is chosen such that it is about twice the length of the largest target in spatial dimensions and, $\mathcal{W}_i$ represents the support for the i th window. The overall metric is then obtained by averaging the computed metric values for each window over all N windows.

A.1.2 Schmieder Weathersby

$$SW_{\text{metric}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2} \tag{A.2}$$

where σ_i^2 is the variance of pixels within the ith window.

A.1.3 Homogeneity

$$H_i = \sqrt{\frac{1}{W}\sum_{j \in \mathcal{W}_i}(I_j - \bar{I}_i)^2}\,, \tag{A.3}$$

where

$$\bar{I}_i = \frac{1}{W}\sum_{j \in \mathcal{W}_i} I_i \tag{A.4}$$

$$Homogeneity_{\text{metric}} = \frac{1}{N}\sum_{i=1}^{N} H_i \tag{A.5}$$

A.1.4 Energy

$$Energy_i = \sum_{j=0}^{GL-1}(P_i[j])^2 \tag{A.6}$$

$$Energy_{\text{metric}} = \frac{1}{N}\sum_{i=1}^{N} Energy_i \tag{A.7}$$

A.1.5 Entropy

$$Entropy_i = -\sum_{j=0}^{GL-1}(P_i[j])\log_2(P_i[j]) \tag{A.8}$$

$$Entropy_{\text{metric}} = \frac{1}{N}\sum_{i=1}^{N} Entropy_i \tag{A.9}$$

In both definitions in A.1.4 and A.1.5, GL is the defined number of gray-level intensity values (typically 256) in the image and, P_i is the histogram of the intensities of the pixels in the ith window.

A.1.6 Target Interference Ratio

$$TIR_i = |\mu_{\text{target}} - \mu_{\text{background}}|/\sigma_{\text{background}} \tag{A.10}$$

$$TIR_{\text{metric}} = \frac{1}{N}\sum_{i=1}^{N} TIR_i \tag{A.11}$$

where μ_{target} is the mean of the intensity values in a window of about the same size as the target, and $\mu_{background}$ and $\sigma_{background}$ are the mean and standard deviation of the target background. The target background is defined as the window centered around the target but twice the dimensions of the target. In this case, the size of the target background determines the value of N.

A.1.7 Outlier/Edge

$$Edge_i = \text{Cardinality of } \{j : |I_j - \bar{I}_i| > \bar{I}_i/4\} \tag{A.12}$$

where $j \in \mathcal{W}_i$ and $\bar{I}_i = \frac{1}{W}\sum_{j\in\mathcal{W}_i} I_i$

$$Edge_{\text{metric}} = \frac{1}{N}\sum_{i=1}^{N} Edge_i \tag{A.13}$$

A.1.8 FBM Hurst Parameter

$$f_s^x = \frac{1}{D_1 D_2}\sum_{i=1}^{D_1}\sum_{j=1}^{D_2} |I(i+2^s, j) - I(i,j)|^2 \tag{A.14}$$

$$f_s^y = \frac{1}{D_1 D_2}\sum_{i=1}^{D_1}\sum_{j=1}^{D_2} |I(i, j+2^s) - I(i,j)|^2 \tag{A.15}$$

$$f_s = f_s^x + f_s^y \tag{A.16}$$

$$FBM_{\text{metric}} = \text{slope}\,(\frac{1}{2}\log_2 f_s) \tag{A.17}$$

where D_1 and D_2 are the spatial dimensions of a hyperspectral band. f is computed for

$1 : s$, where s is determined as

$$s = \text{floor}((\log(N_s)/\log(2)) - 2)\,, \tag{A.18}$$

where the floor operator rounds the expression to the nearest integer towards minus infinity, and $N_s = \text{minimum}(D_1, D_2)$.

A.1.9 Metrics c and p derived from Gabor filtered images [9]

The Gabor filter we used is bandpass filter with a Gaussian kernel. It is defined as

$$F_{\sigma,\theta}(z) = \exp(-\frac{1}{2\sigma^2}(z_\theta(1)^2 + z_\theta(2)^2)) \exp(-j\frac{2\pi z_\theta(1)}{\sigma}) \tag{A.19}$$

where $\sigma = 4$ denotes the resolution associated with the filter and

$$z_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^2 \tag{A.20}$$

and $\theta \in [0, 2\pi)$ is the filter rotation angle. For a bank of K filters, we obtain $F^{(j)}, j = 1, 2, ..., K$. For a particular rotation angle, the filtered images is obtained by the 2D convolution of the image with the filter

$$I^{(j)} = I * F^{(j)} \tag{A.21}$$

p and c are obtained as:

$$p_{\text{metric}} = \frac{3}{SK(I^{(j)}) - 3} \tag{A.22}$$

$$c_{\text{metric}} = \frac{SV(I^{(j)})}{p} \tag{A.23}$$

where SK and SV are the sample kurtosis and the sample variance of the Gabor filtered image respectively.

A.2 Metric derived from band information content

A.2.1 Band correlation

$$BC_{\text{metric}} = \sum_{i=1}^{L} \sum_{j=i+1}^{L} CC(b_i, b_j) \tag{A.24}$$

where L is the number of hyperspectral bands, and CC denotes the correlation coefficient between bands b_i and b_j.

A.3 Anomaly detectors

A.3.1 Dot product

$$DP_{\text{metric}} = \frac{1}{T} \sum_{i=1}^{T} (1 - (\frac{1}{n} \sum_{j=1}^{n} (x_i/|x_i|) \cdot (y_{ij}/|y_{ij}|))) \tag{A.25}$$

where x_i is the pixel vector under test and y_{ij} are the pixel vectors surrounding the vector under test, all of length L. Typically, $n = 4$, and the test pixels are located at the vertices of a square with the test pixel as center and sides of length typically equal to 3 pixels. T is the total number of pixels in the spatial dimensions minus the pixels at the edges.

A.3.2 Kullback-Leibler

$$KL_{ij} = \sum_{k=1}^{GL-1} P_k(x_i) \times \log(\frac{P_k(x_i)}{P_k(y_{ij})}) \tag{A.26}$$

where $P(x_i)$ is the histogram of the vector under test, $P(y_{ij})$ is the histogram of one of the surrounding pixels and $GL = 256$ is the number of gray-levels for the histogram definition. The above is thus the Kullback-Leibler distance between these two pixel

vectors. The arrangement of the surrounding pixels is the same as in A.3.1. and the metric value for a particular test pixel is obtained by averaging this distance over the $n = 4$ surrounding pixels. The overall Kullback-Leibler metric is obtained by averaging each pixel metric value over all T tested pixels, where T is also as described in A.3.1.

$$KL_{\text{metric}} = \frac{1}{T}\sum_{i=1}^{T}(\frac{1}{n}\sum_{j=1}^{n} KL_{ij}) \tag{A.27}$$

A.4 Metrics derived from the Gray Level Co-occurrence Matrix

Given intensity values $I(i, j, l)$, where (i, j) is the spatial location and l is the band location, and the number of gray-levels is GL (typically 256) the GLC matrix G is obtained thus:

for $t = 1 : T$

$$m = I(i_t, j_t, b_t);\ \ n = I(i'_t, j'_t, b'_t)$$

$$G(m, n) = G(m, n) + 1$$

end

T is the total number of samples used. The offsets in the 3 dimensions are $(i_t - i'_t, j_t - j'_t, l_t - l'_t)$. $G \in \mathbb{R}^2$ has size 256×256.

The derived metrics are:

$$GLCM\_Imax. = 1/max(max(G)) \tag{A.28}$$

$$GLCM\_Energy = \sum_m \sum_n G_{mn}^2 \tag{A.29}$$

$$GLCM\_Entropy = -\sum_m \sum_n G_{mn} \log(G_{mn}) \tag{A.30}$$

$$GLCM\_Contrast = \sum_m \sum_n G_{mn}(m - n)^2 \tag{A.31}$$

$$GLCM\_Homogeneity = \sum_{m} \sum_{n} G_{mn}/(m-n) \tag{A.32}$$

REFERENCES

[1] ARNOLD, P. S., BROWN, S. D., AND SCHOTT, J. R. Hyperspectral simulation of chemical weapon dispersal patterns using dirsig. In *SPIE Proceedings* (2000), vol. 4029, pp. 288–299.

[2] ASMAR, D. C., ELSHAMLI, A., AND AREIBI, S. A comparative assessment of aco algorithms within a tsp environment. *Dynamics of Continuous, Discrete, and Impulsive Systems - Series B - Applications and Algorithms 1* (2005), 462–467.

[3] BACK, T. *Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996.

[4] BALLINGER, D. Space-based hyperspectral imaging and its role in the commercial world. In *Proceedings of IEEE Aerospace Conference* (2001), vol. 2, pp. 915–923.

[5] BAUTISTA, P., ABE, T., YAMAGUCHI, M., YAGI, Y., AND OHYAMA, N. Digital staining for multispectral images of pathological tissue specimens based on combined classification of spectral transmittance. *Journal of Computerized Medical Imaging and Graphics 29(8)* (2003), 649–657.

[6] BERK, A. Moderate spectral resolution atmospheric transsmittance algorithm and computer model documentation. `www.vs.afrl.af.mil/ProductLines/IR-Clutter/modtran4`.

[7] BONGIOVI, R. P., HACKWELL, J. A., AND HAYBURST, T. L. Airborne lwir hyperspectral measurements of military vehicles. In *Proceedings of IEEE Aerospace Application Conference* (1996), vol. 3, pp. 121–135.

[8] BRITTON, D., SMITH, M., AND MERSEREAU, R. Generalized gaussian decompositions for image analysis and synthesis. In *Proceedings of the Systematics, Cybernetics, and Informatics (SCI)* (Orlando, FL, July 2000).

[9] BROWN, J. H. Strategic high altitude atmospheric radiance code documention. `www.vs.afrl.af.mil/ProductLines/IR-Clutter/sig.aspx`.

[10] BROWN, S. D. Digital imaging and remote sensing image generation model documention. `http://dirsig.cis.rit.edu/docs`.

[11] CARO, G. D., DUCATELLE, F., AND GAMBARDELLA, L. M. Anthocnet: an adaptive nature inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications 16(5)* (2005), 443–455.

[12] CARPIN, S., AND PILLONETTO, G. Motion planning using adaptive random walks. *IEEE Transactions on Robotics 21(1)* (2005), 129–136.

[13] COOK, R. L. Stochastic sampling in computer graphics. *ACM Transactions on Graphics 5(1)* (1986), 51–72.

[14] DENEUBORG, J. L., AND GOSS, S. Collective patterns and decision making. *Ethology, Ecology and Evolution 76* (1989), 579–581.

[15] DIPPE, M. A. Z., AND WOLD, E. H. Antialiasing through stochastic sampling. In *Proceedings of SIGGRAPH* (San Francisco, 1985), vol. 19(3), pp. 69–78.

[16] DJURIC, P., HUANG, Y., AND GHIRMAI, T. Perfect sampling: a review and applications to signal processing. *IEEE Transactions on Signal Processing 50(2)* (2002), 345–356.

[17] DORIGO, M., BIRATTARI, M., AND STIITZLE, T. Ant colony optimization: artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine* (2006), 28–39.

[18] DORIGO, M., AND GANBARDELLA, L. Ant colony system: a cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computing 1(1)* (1997), 53–66.

[19] DORIGO, M., MANIEZZO, V., AND COLORNI, A. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B 26(1)* (1996), 1–13.

[20] DORIGO, M., MANIEZZO, V., AND COLORNI, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics - Part B 26(1)* (1996), 29–41.

[21] ELDAR, Y. Irregular image sampling using the voronoi diagram. Master's thesis, Technicon-IIT, Israel, 1992.

[22] ELDAR, Y., LINDENBAUM, M., PORAT, M., AND ZEEVI, Y. Y. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing 6(9)* (1997), 1305–1315.

[23] FADIRAN, O. O., AND KAPLAN, L. M. Clutter complexity analysis of hyperspectral bands. In *Proceedings of the Thirty-Sixth I.E.E.E. Southeastern Symposium* (Atlanta, 2004), pp. 531–535.

[24] FADIRAN, O. O., AND MOLNAR, P. Adaptive sampling based on frequency distribution of function values. In *Proceedings of the SIAM Conference on Imaging Science* (May 2006).

[25] FADIRAN, O. O., MOLNAR, P., AND KAPLAN, L. M. Towards quantifying clutter in hyperspectral infrared images. In *Proceedings of the IASTED Conference on Circuits, Signals and Systems* (San Francisco, 2005).

[26] FADIRAN, O. O., MOLNAR, P., AND KAPLAN, L. M. Adaptive sampling via histogram equalization using an active-walker model. In *Proceedings of the IEEE International Conference on Information Science* (July 2006), pp. 424–432.

[27] FADIRAN, O. O., MOLNAR, P., AND KAPLAN, L. M. A statistical approach to quantifying clutter in hyperspectral infrared images. In *Proceedings of the IEEE Aerospace Conference* (Big Sky, March 2006), pp. CD–ROM.

[28] FOGEL, D. B. Evolutionary programming. In *Proceedings of the 5th annual conference on evolutionary programming* (1996).

[29] FREIMUTH, R. D., AND LAM, L. Active walker models for filamentarty growth patterns. In *Modeling Complex Phenomena* (1992), L. Lam and V. Naroditsky, Eds., New Tork: Springer, pp. 302–313.

[30] GORGES-SCHLEUTER, M. A comparative study of local and global selection in evolution strategies. In *Proceedings of the PPSN Conference* (1998), pp. 367–377.

[31] GOSS, S., ARON, S., DENEUBORG, J. L., AND PASTEELS, J. M. Self-organized shortcuts in the argentine ant. *Naturwissenschaften 1* (1989), 295–311.

[32] GOSS, S., BECKERS, R., DENEUBORG, J. L., ARON, S., AND PASTEELS, J. M. How trail laying and trail following can solve foraging problems for ant colonies. *Behavioural Mechanisms of Food Selection G20* (1990).

[33] GRAVEL, M., PRICE, W. L., AND GAGNE, C. Scheduling continuous casting of aluminium using a multiple objective ant colony optimization metaheuristic. *European Journal on Operations Research 143* (2002), 218–229.

[34] GRAY, R. M. *Entropy and information theory.* Springer-Verlag, 1990.

[35] HARALICK, R. M. Statistical and structural approaches to texture. In *Proceedings of the 4th Joint Conference on Pattern Recognition* (1978), pp. 45–69.

[36] HAUTA-KASARI, M., PARKKINEN, J., JAAKELAINEN, T., AND LENZ, R. Generalized co-occurrence matrix for multispectral texture analysis. In *Proceedings of the 13th International Conference on Pattern Recognition* (1996), vol. 2, pp. 782–789.

[37] HELBING, D., KELTSCH, J., AND MOLNR, P. Modelling the evolution of human trail systems. *Nature 388* (1997), 47–50.

[38] HELBING, D., SCHWEITZER, F., KELTSCH, J., AND MOLNR, P. Active walker model for the formation of human and animal trail systems. *Physical Review E 56* (1997), 2527–2533.

[39] HOLLAND, O., AND MELHUISH, C. Stigmergy, self-organization, and sorting in collective robotics. *MIT Press Journal 5(2)* (1999), 173–202.

[40] HOLLDOBLER, B., AND MOGLICH, M. *The ants.* Belknap, 1990.

[41] HUANG, Y., AND DJURIC, P. M. Variable selection by perfect sampling. *EURASIP Journal on Applied Signal Processing 2002(1)* (2002), 38–45.

[42] HUNG, K. L., AND CHANG, C. C. New irregular sampling coding method for transmitting images progressively. In *Proceedings of the IEEE Conference on Vision, Images and Signal Processing* (February 2003), pp. 44–50.

[43] JAIN, A. K. *Fundamentals of digital image processing*. Prentice Hall, 1989.

[44] JOHNSON, A., WINDESHEIM, E., AND BROCKHAUS, J. Hyperspectral imagery for trafficability analysis. In *Proceedings of IEEE Aerospace Conference* (1998), vol. 2, pp. 21–35.

[45] KAPLAN, L., NAMUDURI, K., DAVIES, M., NASRABADI, N., CHAN, L., BRITTON, D., SMITH, M., AND MERSEREAU, R. Development and analysis of a clutter complexity measure for ATR characterization. In *Proc. of the Fifth Annual ARL Federated Laboratory Symposium* (College Park, MD, Mar. 2001), pp. 195–199.

[46] KAYSER, D. R., ABERLE, L. K., POCHY, R. D., AND LAM, L. Active walker models: tracks and landscapes. *Physica A 191* (1992), 17–24.

[47] KOZA, J. R. *Genetic programming: on the programming of computers by means of natural selection.* MIT Press, 1992.

[48] LAM, L. Active walker models for complex systems. *Journal of Chaos, Solitons and Fractals 6* (1995), 267–285.

[49] LAM, L., AND POCHY, R. Active-walker models: growth and form in nonequilibrum systems. *Computers in Physics 7* (1993), 534–541.

[50] LANDGREBE, D. A. *Signal Theory Methods in Multispectral Remote Sensing.* Wiley-Interscience, 2003.

[51] LANTERMAN, A., O'SULLIVAN, J., AND MILLER, M. Kullback-Leibler distances for quantifying clutter and models. *Optical Engineering 38* (Dec. 1999), 2134–2146.

[52] LEE, M. E., REDNER, R. A., AND USELTON, S. P. Statistically optimized sampling for distributed ray tracing. In *Proceedings of SIGGRAPH* (San Francisco, 1985), vol. 19(3), pp. 61–67.

[53] LEE, M. E., REDNER, R. A., AND USELTON, S. P. A statistical method for adaptive stochastic sampling. In *Proceedings of Eurographics Conference and Exhibition* (Lisbon, 1986), vol. 11(2), pp. 157–162.

[54] LENTILUCCI, E. J., BROWN, S. D., SCHOTT, J. R., AND RAQUENO, R. V. Multispectral simulation environment for modelling low-light-level sensor systems. In *SPIE Proceedings* (1998), vol. 3434, pp. 10–19.

[55] MANIEZZO, V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS Journal on Computing 11(4)* (1999), 358–369.

[56] MARKS, R. J. *Introduction to Shannon sampling and interpolation theory.* Springer Verlag, 1991.

[57] MCWHORTER, L. T., AND SCHARF, L. L. Adaptive matched subspace detectors and adaptive coherence estimators. In *Proceedings of the 30th Asilomar Conference on Signals, Systems, and Computers* (1996), pp. 1114–1117.

[58] MEITZLER, T., GERHART, G., AND SINGH, H. A relative clutter metric. *IEEE Trans. on Aerospace and Electronic Systems 34* (July 1998), 968–976.

[59] MHLENBEIN, H., GORGES-SCHLEUTER, M., AND KRMER, O. Evolution algorithms in combinatorial optimization. *Parallel Computing 7* (1988), 65–85.

[60] MITCHELL, D. P. Generating antialiased images at low sampling densities. In *Proceedings of SIGGRAPH* (1987), vol. 21(4), pp. 65–72.

[61] MITCHELL, M. *An introduction to genetic algorithms.* MIT Press, 1998.

[62] NAMUDURI, K. R., BOUYOUCEF, K. K., AND KAPLAN, L. M. Image metrics for clutter characterization. In *Proceedings of 2000 IEEE International Conference on Image Processing* (2000), vol. 2, pp. 467–470.

[63] NORRIS, J. R. *Markov chains.* Cambridge University Press, 1999.

[64] PAINTER, J., AND SLOAN, K. Antialiased ray tracing by adaptive progressive refinement. In *Proceedings of SIGGRAPH* (1989), vol. 23(3), pp. 281–288.

[65] PANUSOPONE, K., CHEEVASUVIT, F., AND RAO, K. R. Adaptive subsampling for image compression. In *Proceedings of the IEEE ASILOMAR Conference* (1996), pp. 239–243.

[66] PETERS, R. A., AND STRICKLAND, R. N. Image complexity metrics for automatic target recognizers. In *Proceedings of Automatic Target Recognizer System and Technology System* (Silver Spring, 1990).

[67] PROAKIS, J. G., AND MANOLAKIS, D. G. *Digital Signal Processing: Principles, Algorithms, and Applications.* Pearson Prentice Hall, 1996 (Fourth edition).

[68] REED, I., AND YU, X. Adaptive multi-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Trans. on Acoustics, Speech, and Signal Processing 38* (Oct. 1990), 1760–1770.

[69] REIMANN, M., DOERNER, K., AND HARTL, R. F. D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers and Operations Research 31(4)* (2004), 563–591.

[70] RICHTSMEIER, S. C., SUNDBERG, R. L., BERK, A., AND ADLER-GOLDEN, S. M. Full-spectrum scene simulation. In *SPIE proceedings* (2004), vol. 5425, pp. 530–537.

[71] RIGAU, J., FEIXAS, M., AND SBERT, M. Entropy-based adaptive sampling. In *Proceedings of Graphics Interface Conference* (Halifax, June 2003).

[72] RODNICK, J., AND GASPARI, G. *Elements of the random walk.* Cambridge University Press, 2004.

[73] SANDERS, J. S., AND BROWN, S. D. Utilization of dirsig in support of real-time infrared scene generation. In *SPIE Proceedings* (2000), vol. 4029, pp. 278–285.

[74] SCANLAN, N. W., SCHOTT, J. R., AND BROWN, S. D. Performance analysis of improved methodology for incorporation of spatial/spectral variability in synthetic hyperspectral imagery. In *SPIE Proceedings* (2003), vol. 5159, pp. 319–330.

[75] SCHIMANSKY-GEIER, L., MIETH, M., ROSE, H., AND MALCHOW, H. Structure formation by active brownian particles. *Physics Letters A 207* (1995), 140.

[76] SCHOTT, J. R., BROWN, S. D., RAQUENO, R. V., GROSS, H. N., AND ROBINSON, G. An advanced synthetic image generation model and its application to

multi/hyperspectral algorithm development. *Canadian Journal of Remore Sensing 15* (1999), 99–111.

[77] SCHOTT, J. R., KUO, S. D., BROWN, S. D., AND RAQUENO, R. V. Prediction of observed image spectra using synthetic image generation models. In *SPIE Proceedings* (1997), vol. 3118, pp. 81–93.

[78] SCHWEITZER, F., AND SCHIMANSKY-GEIER, L. Clustering of active walkers in a two-component system. *Physica A 206* (1994), 359–379.

[79] SCHWEITZER, F., AND SCHIMANSKY-GEIER, L. Clustering of active walkers: phase transition from local interaction. In *Fluctuations and Order: The New Synthesis* (1996), M. Millonas, Ed., New Tork: Springer, pp. 293–305.

[80] SHI, M., AND HEALEY, G. E. Three-dimensional hyperspectral texture recognition using multiband correlation models. In *SPIE Proceedings* (2003), vol. 5093, pp. 678–685.

[81] SHIRVAIKAR, M. V., AND TRIVEDI, M. M. Developing a texture-based image clutter measure for object detection. *Optical Engineering 31* (Dec. 1992), 2628–2639.

[82] SIEGMUND, D. *Elements of information theory*. Wiley Interscience, 1991.

[83] SIMS, S. Putting ATR performance on an equal basis: The measurement of knowledge based distortion and relevant clutter. In *Proc. of SPIE* (July 1999), vol. 3810, pp. 118–128.

[84] SRINIVAS, N., AND DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation 2(3)* (1994), 221–248.

[85] SRIVASTAVA, A., LIU, X., AND GRENANDER, U. Universal analytical forms for modeling image probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24(9)* (2002), 1200–1214.

[86] STUTZLE, T., AND HOOS, H. H. Max-min ant system. *Future Generation Computer Systems 16(8)* (2000), 889–914.

[87] SVENSON, P., AND SIDENBLADH, H. Determining possible avenues of approach using ants. In *Proceedings of the 6th International Conference on Information Fusion* (2003), pp. 1110–1117.

[88] TAMSTORF, R., AND JENSEN, H. W. Adaptive sampling and bias estimation in path tracing. In *Proceedings of the 8th Eurographics Workshop on Rendering* (St. Etienne, 1997), pp. 285–295.

[89] TAXT, T., AND LUNDERVOLD, A. Multispectral analysis of the brain using magnetic resonance imaging. *IEEE Transactions on Medical Imaging 13(3)* (1994), 470–481.

[90] VO-DINH, T., CULLUM, B., AND KASILI, P. Development of a multi-spectral imaging system for medical applications. *Journal of Physics D: Applied Physics 36(14)* (2003), 1663–1668.

[91] WALDMAN, G., WOOTTON, J., HOBSON, G., AND LUETKEMEYER, K. A normalized clutter measure for images. *Journal of Computer Vision, Graphics, and Image Processing 42* (1998), 137–156.

[92] WANG, Z., BOVIC, A. C., SHEIKH, H. R., AND SIMONCELLI, E. E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image processing 13(4)* (2004), 600–612.

[93] WINDHAM, R. W., WILLIAM, R., LAWRENCE, C. K., PARK, B., SMITH, P. D., AND POOLE, G. Analysis of reflectance spectra from hyperspectral images of poultry carcasses for fecal and ingesta detection. In *SPIE proceedings* (2002), vol. 4816, pp. 317–324.

[94] ZHU, S., LANTERMAN, A., AND MILLER, M. Clutter modeling and performance analysis in automatic target recognition. In *Proceedings of Workshop on Classification and Detection of Difficult Targets* (Redstone Arsenal, 1998), pp. 477–496.